2020 Vulnerability Intelligence Report

rapid7.com/research/report/vulnerability-intelligence-report/

- Home
- Research

Caitlin Condon, Software Engineering Manager at Rapid7
Spencer McIntyre, Lead Security Researcher at Rapid7
William Vu, Senior Security Researcher at Rapid7

Introduction

Security, IT, and other teams tasked with vulnerability management and risk reduction frequently operate in high-urgency, high-stakes environments where informed decision-making hinges on the ability to quickly separate signal from a sea of perpetual noise. When a new potential threat emerges, information security professionals often find themselves needing to translate vague descriptions and untested research artifacts into actionable intelligence for their own particular risk models.

Rapid7 researchers analyze thousands of vulnerabilities each year to understand root causes, dispel misconceptions, and share information on why certain flaws are more likely to be exploited than others. This report examines 50 vulnerabilities from the 2020 calendar year in order to highlight exploitation trends, explore attacker use cases, and offer a framework for understanding new security threats as they arise. Our aim is to contextualize the vulnerabilities that introduce serious risk to a wide range of organizations—and those that probably don't. We have also included a practical applications section with guidance for defenders.

This is not a threat intelligence report, though we do make use of cyber threat intelligence (CTI) terminology within it. There are many fine purveyors of threat intelligence who do a first-rate job of mapping motives and tactics, techniques, and procedures (TTPs) to specific threat actors. Traditional CTI methodologies and elements, including artifacts like indicators of compromise (IOCs), are outside the scope of this paper.

There is a substantial amount of data incorporated into the sections below, including a number of overarching metadata types Rapid7 researchers have defined and normalized. It's all important, but don't let it fool you: This is very much a report about instinct—the hunches that drive security researchers and exploit developers—with a dash of history to boot.

Executive Summary

The volume of published vulnerabilities has grown significantly over the past five years. 2017 experienced a 127% increase in CVE-named vulnerabilities over 2016, and each year of CVEs since then has topped its predecessor in sheer volume. By the time the year came to a close, 2020 racked up 18,362 vulnerabilities, which represented a 6% increase over 2019 and a dizzying 185% rise from five years ago. December 2020's revelations about nation state-backed campaigns that compromised more than half a dozen U.S. government agencies and thousands of organizations globally capped a turbulent year of high-priority security flaws, some of the most severe of which occurred in internet-facing technologies critical to protecting a newly remote workforce.

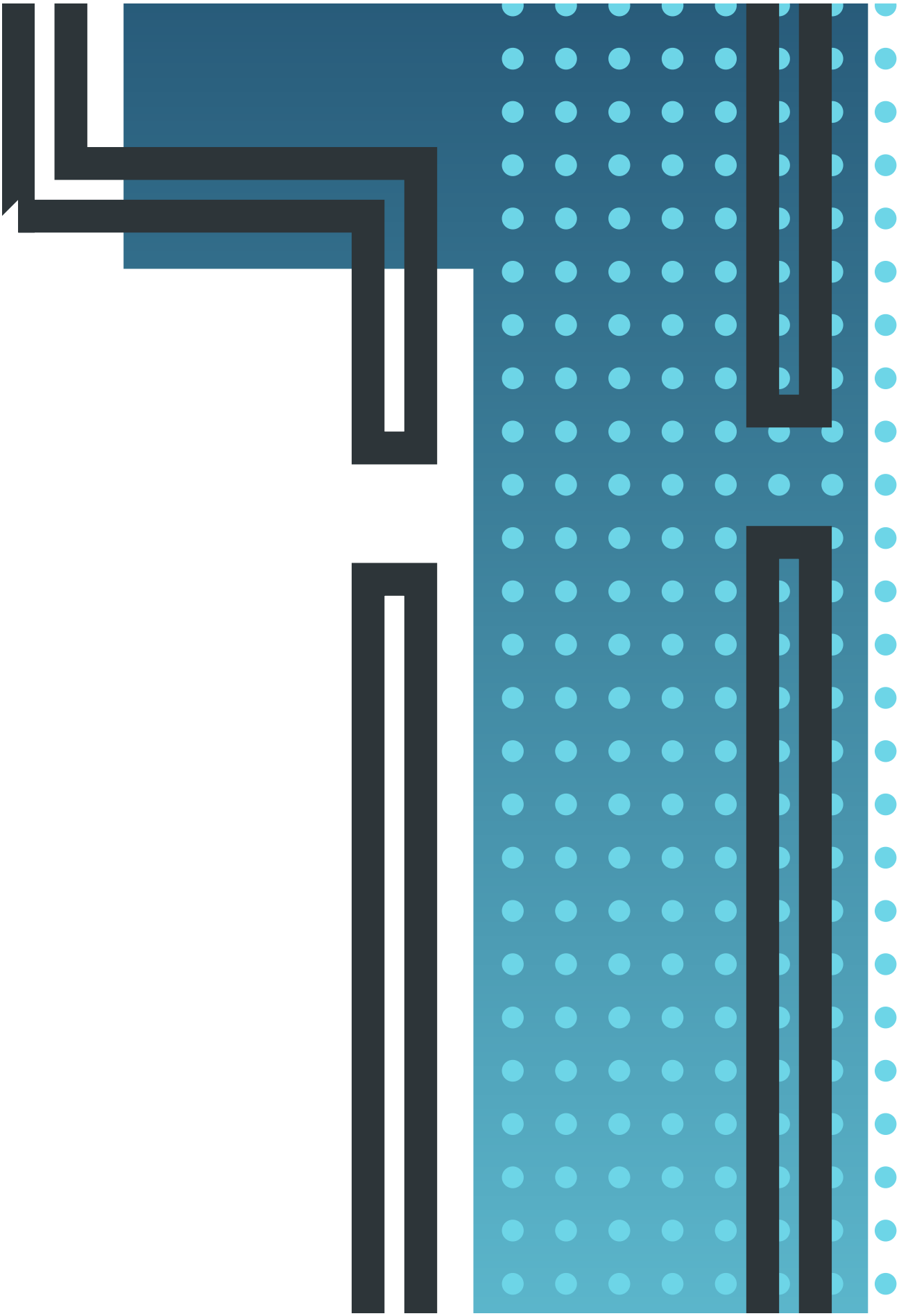
Of course, not all of 2020's security flaws stood shoulder to shoulder—even those given perfect 10 severity scores.

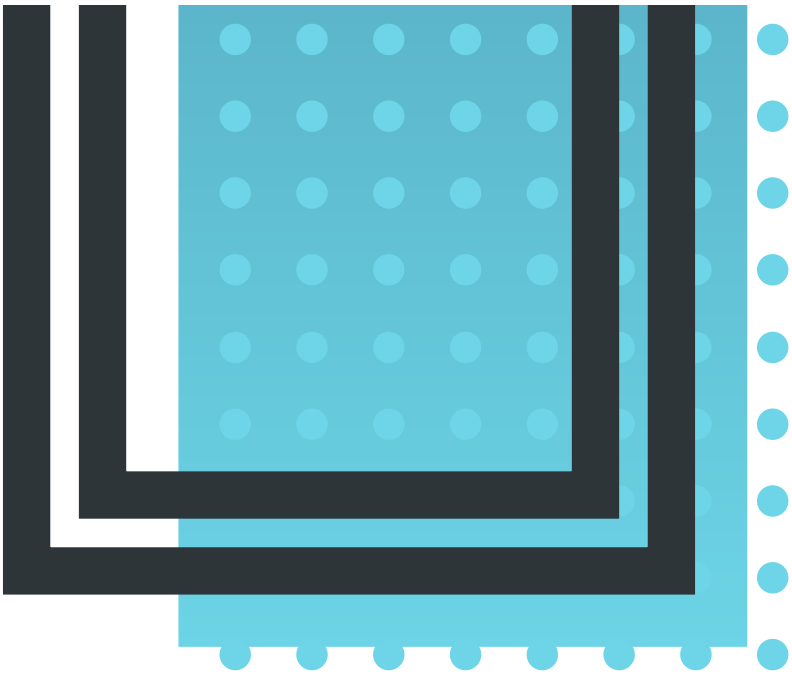
We look at 50 distinct CVEs in this report: 30 of these vulnerabilities were exploited in the wild in 2020, either broadly in indiscriminate attacks or in smaller-scale, targeted attacks. The remaining 20 vulnerabilities are not known to be actively exploited and are classified as impending threats because of their value to attackers, the discoverability of technical details, and/or the general availability of exploit code. Of course, the absence of evidence doesn't mean these vulnerabilities haven't been exploited—only that no exploitation has been publicly disclosed.

2020 findings include:

- **In this report, we examine 14 vulnerabilities that became widespread threats and posed substantial risks to organizations of all sizes in 2020.** You will undoubtedly recognize many of these widely exploited vulnerabilities, which include WebLogic Server and ManageEngine zero-days in addition to the likes of RECON and Zerologon. The 14 CVEs in our widespread threat list were exploited by a wide range of malicious parties, from state-sponsored threat actors to run-of-the-mill "commodity" attackers.
- **Critical vulnerabilities featured prominently in security news headlines in 2020, but some flaws were much more exploitable than others.** Deserialization and improper access control vulnerabilities were the root cause of some of the most widespread threats of 2020, accounting for 78% of CVEs in our dataset that were exploited at scale.
- **Some of 2020's most high-impact vulnerabilities occurred in security products that sat in critical and often exposed places in organizations' networks.** Nine of the vulnerabilities we examine in this report functioned as network pivots, providing opportunities for external attackers to gain internal network access by exploiting VPNs, firewalls, or other internet-facing technologies. These flaws are often used in conjunction with local code execution vulnerabilities or network protocol bugs to escalate privileges or move laterally across corporate networks.
- **2020 included multiple vulnerability suites that disproportionately affected operational technology (OT) and Internet of Things (IoT) implementations of low-level software libraries.** While these vulnerability groups garnered quite a lot of attention, Rapid7 and SCADAfence research teams independently determined their exploitability to be much lower than some other vulnerabilities affecting the OT ecosystem. Vendor and supply chain fragmentation further underscored the need for long-term visibility across OT networks.
- **Defenders had to contend with a steady stream of patch bypasses in 2020, which disrupted remediation cycles and introduced risk that was more likely to be missed.** Our vulnerability dataset includes nine patch bypasses or incomplete patches, seven of which circumvent fixes for known-exploited or high-value parent vulnerabilities. One trend we observed is that it's common for software vendors to mitigate disclosed attack chains rather than addressing the root cause of bugs reported to them; this may partially account for the prevalence of patch bypasses last year.

Big Picture: 2020 Threats



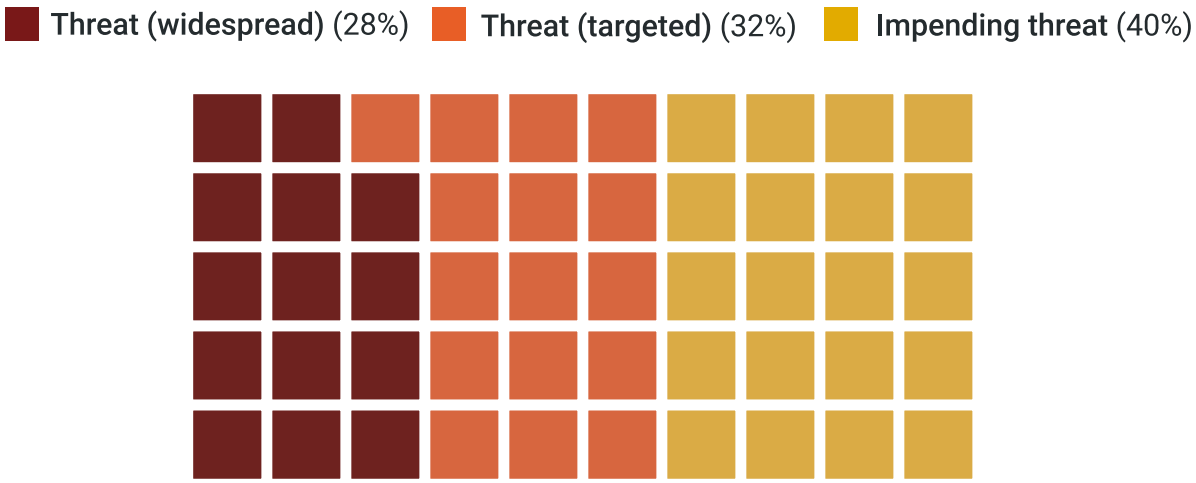


Big Picture: 2020 Threats Actively Exploited and Other Significant CVEs

In an interesting twist, 2020 marked the advent of zero-day disclosures and public threat intelligence coming directly from the U.S. National Security Agency (as well as from more traditional sources like CISA and country CERTs). This nascent practice, no doubt intended to accelerate patching as much as to demonstrate a new commitment to transparency, has contributed to a heightened sense of urgency—and occasionally alarm—that is amplified by security news media reporting (rightfully) on intelligence agency alerts. It is difficult to discuss the threat landscape of the past year without acknowledging the role U.S. agencies have played in shaping perceptions and influencing priorities, particularly since news headlines on these disclosures are likely to reach executives and business stakeholders as well as security practitioner audiences.

2020 Vulnerabilities by Threat Status

n = 50



We make a distinction in our data between vulnerabilities that see widespread, indiscriminate exploitation (widespread threats) and those that, based on available evidence, have been used in highly targeted attacks (targeted threats). Almost all of the targeted threats in our dataset were reported by a single credible source as having been exploited, with no further public reports of exploitation thereafter. This includes many of the flaws in our targeted threat category that were zero-days when they were disclosed.

Conversely, CVEs in our widespread threat category have typically been exploited by a range of malicious actors, from botnet controllers to APTs to low-skilled attackers throwing public proof-of-concept (PoC) code against internet-exposed hosts. In other words, if we have classified a threat as widespread, that classification is based on the volume of attacks rather than who specifically is doing the attacking. When applying learnings from 2020 to the current year and beyond, organizations should expect to conduct incident response investigations that look for IOCs and suspicious activity during widespread threat events in addition to activating emergency patching protocols.

What is a threat?

When there is an adversary with the intent, capability, and opportunity, a threat exists. When two or more of these elements are present (e.g., intent and capability, but no opportunity), we call it an impending threat, because there is just one missing piece before it becomes a true threat. When there is just one element present (e.g., an opportunity in the form of a software vulnerability), we call it a potential threat. There is the potential for it to turn into a true threat, although there are additional components that need to come to fruition before it has a real impact to most organizations.

Widespread Threats

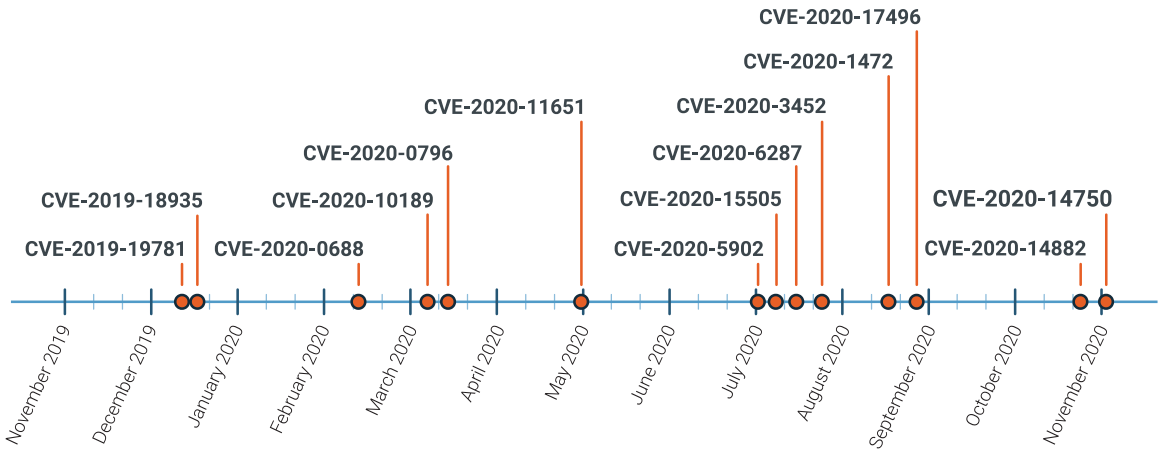
Fourteen CVEs posed widespread threats to organizations in 2020 and are likely to stalk unpatched systems well into 2021. The impact of successful exploitation for each of these vulnerabilities is high; most allow for remote code execution (RCE) at a minimum, but several allow unauthenticated, remote attackers to take over infrastructure or gain access to internal networks through exploitation of vulnerable internet-facing systems or interfaces. It's worth noting that two of the entries in this category—Zoho ManageEngine CVE-2020-10189 and vBulletin CVE-2020-17496—were released into the wild as zero-days with stable exploit code before the vendors made patches available.

CVE	CVSS v3	Threat Status	Attacker Utility	Vulnerability Class
CVE-2019-18935 Telerik UI RadAsyncUpload .NET Deserialization	9.8	• Widespread Threat Exploited in the wild	Remote code execution	Deserialization / .NET
CVE-2019-19781 Citrix NetScaler ADC/Gateway/SD-WAN Arbitrary Code Execution	9.8	• Widespread Threat Exploited in the wild	Network pivot	Improper Access Control
CVE-2020-0688 Microsoft Exchange Server Static Validation Key Remote Code Execution	8.8	• Widespread Threat Exploited in the wild	Remote code execution	Deserialization / .NET
CVE-2020-0796 Windows SMBv3 Server Remote Code Execution "SMBGhost"	10	• Widespread Threat Exploited in the wild	Remote code execution	Memory Corruption
CVE-2020-10189 Zoho ManageEngine Remote Code Execution	9.8	• Widespread Threat Exploited in the wild (released as 0day)	Network infrastructure compromise	Deserialization / Java

CVE-2020-11651 SaltStack Salt Remote Code Execution	9.8	• Widespread Threat Exploited in the wild	Network infrastructure compromise	Improper Access Control
CVE-2020-1472 NetLogon Elevation of Privilege "Zerologon"	10	• Widespread Threat Exploited in the wild	Lateral movement	Improper Access Control
CVE-2020-14750 WebLogic Unauthenticated Remote Code Execution Patch Bypass	9.8	• Widespread Threat Exploited in the wild	Remote code execution	Improper Access Control
CVE-2020-14882 Oracle WebLogic Server Remote Code Execution	9.8	• Widespread Threat Exploited in the wild	Remote code execution	Improper Access Control
CVE-2020-15505 MobileIron Core and Connector Remote Code Execution	9.8	• Widespread Threat Exploited in the wild	Remote code execution	Deserialization / Java
CVE-2020-17496 vBulletin subWidgets data RCE	9.8	• Widespread Threat Exploited in the wild (released as 0day)	Remote code execution	Injection
CVE-2020-3452 Cisco ASA/FTD Web Services Read-Only Path Traversal Vulnerability	7.5	• Widespread Threat Exploited in the wild	File enumeration	Injection
CVE-2020-5902 F5 Big-IP TMUI Remote Code Execution	10	• Widespread Threat Exploited in the wild	Network pivot	Improper Access Control

CVE-2020-6287 SAP NetWeaver AS Java "RECON"	10	• Widespread Threat Exploited in the wild	Remote code execution	Improper Access Control

Disclosure dates of widely exploited 2020 vulnerabilities



2020 opened with widespread exploitation of Citrix NetScaler and Telerik UI. Zoho ManageEngine and SaltStack Salt followed suit in the spring, while the second half of the year was punctuated by internet-scale exploitation of severe vulnerabilities in F5 BIG-IP, Microsoft NetLogon Remote Protocol, and Oracle WebLogic Server. The combination of an authentication bypass and a remote code execution flaw in MobileIron's Mobile Device Management (MDM) solution also offered a glimpse at how mobile device management software may prove a potent attack vector in the future, even after the remote workforce has begun to transition back to the office.

There are three outliers in this group that merit closer inspection. The first is CVE-2020-0796, or "SMBGhost," a remote code execution vulnerability in Microsoft's Server Message Block (SMB) 3.1.1 protocol that the company inadvertently published in their March 2020 Patch Tuesday advisory before a patch had been released. An out-of-band security update came within 24 hours of the slip, but not before the security community had likened the flaw to MS17-010, the SMB vulnerability suite exploited as part of the 2017 WannaCry and NotPetya attacks that cost global organizations hundreds of millions of dollars.

Network defenders feared the vulnerability would be widely exploited and wormed, but no worms have materialized, at least thus far. In fact, while public proof-of-concept code and an errant botnet took aim at SMBGhost, it's a bit difficult to argue that the exploits fully arrived, either: Modern memory protections meant that even though a few local exploits made their way to market, remote code execution remains nontrivial (mass migration to remote work may actually have helped limit the vuln's impact). Even the Lemon_Duck botnet appeared to

give up on its SMBGhost exploit module after a trial run. We've included CVE-2020-0796 in our list of widespread threats for the sake of methodological consistency, but it's an exception instead of the rule.

Another exception is CVE-2020-3452, a read-only path traversal flaw in Cisco's Adaptive Security Appliance (ASA) and Firepower Threat Defense (FTD) products that allowed remote attackers to enumerate files on target devices. The anomaly here is value rather than exploitability—although a PoC was quickly published and, to quote Microsoft's Kevin Beaumont, "mass spammed across [the] internet," exploitation offered attackers no useful access at all. CVE-2020-3452 is also the only vulnerability from among our sample of widespread threats for which Metasploit opted not to prioritize an exploit.

Finally, if there *were* a 2020 Microsoft vulnerability that rivaled MS17-010 for severity and catastrophic impact to networked systems, the title would go to "Zerologon" (CVE-2020-1472), a cryptographic flaw in Microsoft's Netlogon Remote Protocol (MS-NRPC) that allows attackers to bypass authentication and compromise Windows Servers running as domain controllers, paving the way for complete takeover of Active Directory domains. The rare CVSS-10 privilege escalation vulnerability in Microsoft's August 2020 Patch Tuesday release raised eyebrows, but the advisory was characteristically sparse. Details remained thin until Secura researchers released in-depth analysis mid-September, which was followed by rampant exploitation.

Both Zerologon and the MS17-010 suite of vulnerabilities gave attackers trivially exploitable paths for lateral movement that lend themselves well to ransomware campaigns (including ransomworms). Unlike the 2017 CVEs, which provided for *both* remote and local code execution, successful exploitation of Zerologon requires network access to a domain controller—in other words, an attacker would need either an internet-exposed domain controller or some type of initial access to the network (e.g., a **network pivot**). 2020's threat list included quite a few potential network pivots; we'll come back to this later on.

Targeted Threats

Roughly half of the actively exploited vulnerabilities in our dataset fell into the "targeted threat" category—which, again, does not mean these CVEs haven't been (or won't be) exploited more broadly, only that there is not yet any public evidence of exploitation at scale. In particular, Sophos XG Firewall CVE-2020-12271 and VMware ESXi CVE-2020-3992 below are incredibly attractive targets for attackers due to the considerable access they provide (Sophos has a detailed analysis of the zero-day attack against their XG Firewall here). The same goes for the three Citrix vulnerabilities, which were resolved in one update and saw small-scale exploitation even as they were moving through coordinated disclosure.

CVE	CVSS v3	Threat Status	Attacker Utility	Vulnerability Class

CVE-2020-10148 VMware Fusion Local Privilege Escalation (Incomplete Patch)	9.8	• Targeted Threat Exploited in the wild (reported as 0day)	Network infrastructure compromise	Improper Access Control
CVE-2020-12271 Sophos XG Firewall Unauthenticated SQL Injection	9.8	• Targeted Threat Exploited in the wild (reported as 0day)	Network pivot	Injection
CVE-2020-17087 Windows Kernel Local Privilege Escalation	7.8	• Targeted Threat Exploited in the wild (reported as 0day)	Local code execution	Memory Corruption
CVE-2020-1020 Windows Adobe Font Manager Library Remote Code Execution	8.8	• Targeted Threat Exploited in the wild (reported as 0day)	Remote code execution	Memory Corruption
CVE-2020-4006 VMware Workspace ONE Command Injection	9.1	• Targeted Threat Exploited in the wild (reported as 0day)	Remote code execution	Injection
CVE-2020-14871 Oracle Solaris PAM Remote Code Execution	10	• Targeted Threat Exploited in the wild (reported as 0day)	Remote code execution	Memory Corruption
CVE-2020-15999 Google Chrome FreeType Heap Buffer Overflow	6.5	• Targeted Threat Exploited in the wild (reported as 0day)	Remote code execution	Memory Corruption

CVE-2020-0986 Windows Sandbox Escape via splw64 Untrusted Pointer Dereference	7.8	• Targeted Threat Exploited in the wild (reported as 0day*)	Local code execution	Memory Corruption
CVE-2020-1048 Windows Print Spooler Service Arbitrary File Write "PrintDemon"	7.8	• Targeted Threat Exploited in the wild (reported as 0day)	Local code execution	Improper Access Control
CVE-2020-8195 Citrix NetScaler ADC/Gateway/SD-WAN WANOP Improper Access Control	6.5	• Targeted Threat Exploited in the wild	Network pivot	Improper Access Control
CVE-2020-8193 Citrix NetScaler ADC/Gateway Authentication Bypass	6.5	• Targeted Threat Exploited in the wild	Network pivot	Improper Access Control
CVE-2020-8196 Citrix NetScaler ADC/Gateway/SD-WAN WANOP Improper Access Control	4.3	• Targeted Threat Exploited in the wild	Network pivot	Improper Access Control
CVE-2020-0601 Windows CryptoAPI Spoofing Vulnerability "Curveball"	8.1	• Targeted Threat Exploited in the wild	Remote code execution	Improper Access Control
CVE-2020-3992 VMware ESXi OpenSLP Use-After-Free Remote Code Execution (Incomplete Patch)	9.8	• Targeted Threat Exploited in the wild	Network infrastructure compromise	Memory Corruption
CVE-2020-3118 Cisco IOS XR Software Discovery Protocol Format String Vulnerability "CDPwn"	8.8	• Targeted Threat Exploited in the wild	Remote code execution	Memory Corruption

CVE-2020-1350 Windows DNS Server Remote Code Execution "SigRed""	10	• Targeted Threat Exploited in the wild	Remote code execution	Memory Corruption

*Kaspersky reported CVE-2020-0986 after they had seen in-the-wild attacks targeting the flaw, but said that Microsoft had already prepared a patch by the time they notified them. We still classify this CVE as a zero-day even if Kaspersky was not the only company to observe exploitation.

Eight of the CVEs above—fully half of this group—were reported to the software producers as zero-days that had already seen in-the-wild exploitation: three by Google's Project Zero and/or Threat Analysis Group (CVE-2020-1020, CVE-2020-17087, and CVE-2020-15999), one by the NSA (VMware CVE-2020-4006), and the rest via various security researchers and firms. SolarWinds Orion CVE-2020-10148 is among this group, and it bears mentioning here that it is extremely likely this vulnerability has been more broadly exploited. At time of writing, however, there were no public reports of widespread exploitation—only reports of mass scanning—so we have classified it as a targeted threat until broader exploitation is confirmed.

The CVEs in Windows DNS Server (2020-1350), Windows CryptoAPI (2020-0601), Cisco IOS XR (2020-3118), and VMware Workspace ONE (2020-4006), in addition to all three Citrix vulnerabilities, were included in Q4 2020 publications issued by the NSA on vulnerabilities being exploited by either Russian (VMware CVE-2020-4006) or Chinese (the rest) state-sponsored threat actors. (The NSA was also the entity that disclosed "Curveball" CVE-2020-0601 to Microsoft back in January 2020.) All told, NSA communications were the source material for a significant portion of our 2020 security threat dataset.

Impending Threats

With actively exploited vulnerabilities out of the way, we can turn our attention to some of 2020's notable impending threats. As a brief reminder, an **impending threat** occurs when two elements of the threat triangle are present, but not all three. Since this paper centers on vulnerabilities, the *opportunity* element is present by default. The impending threats we present in this section include either a *capability* (i.e., a mature exploit) or a proxy for measurable *intent.* Quite a few of the vulnerabilities we've classified as targeted threats would have stayed in the impending threat category without the NSA's Q4 2020 bulletins on flaws exploited by state-sponsored actors.

Note: Several of the vulnerabilities in this group have public proof-of-concept (PoC) code available, but for the purposes of this paper, we consider public PoC to be similar to technical details in that it may enable others to build successful exploit chains or improve upon attack

techniques, but it isn't the same as mature code. Proof-of-concept code often has a way to go before it becomes a fully weaponized and documented exploit, if it makes it that far at all. As Marie Antoinette famously pointed out,

> "PoC works on my machine; an exploit works on yours."

CVE	CVSS v3	Threat Status	Attacker Utility	Vulnerability Class
CVE-2020-3950 VMware Fusion Local Privilege Escalation (Incomplete Patch)	7.8	• Impending Threat Exploit Available	Local code execution	Improper Access Control
CVE-2020-25592 SaltStack Salt Authentication Bypass	9.8	• Impending Threat Exploit Available	Network infrastructure compromise	Improper Access Control
CVE-2020-3952 VMware vCenter Server/vmdir Information Disclosure	9.8	• Impending Threat Exploit Available	Network infrastructure compromise	Improper Access Control
CVE-2020-16952 Microsoft SharePoint Authenticated Remote Code Execution	7.8	• Impending Threat Exploit Available	Remote code execution	Deserialization / .NET
CVE-2020-16846 SaltStack Salt Command Injection	9.8	• Impending Threat Exploit Available	Network infrastructure compromise	Injection
CVE-2020-16875 Microsoft Exchange Server DLP Policy Remote Code Execution	8.4	• Impending Threat Exploit Available	Remote code execution	Injection
CVE-2020-5135 SonicWall SonicOS Portal Buffer Overflow	9.4	• Impending Threat High-Value Target	Network pivot	Memory Corruption

CVE-2020-16898 Windows TCP/IP Remote Code Execution "Bad Neighbor"	8.8	• Impending Threat High-Value Target	Remote code execution	Memory Corruption
CVE-2020-8209 Citrix XenMobile Server File Disclosure	7.5	• Impending Threat High-Value Target	File enumeration	Improper Access Control
CVE-2020-3187 Cisco Adaptive Security Appliance/Firepower Threat Defense Path Traversal	9.1	• Impending Threat High-Value Target	File enumeration	Improper Access Control
CVE-2020-2021 Palo Alto Networks PAN-OS SAML Authentication Bypass and Remote Code Execution	10	• Impending Threat High-Value Target	Network pivot	Improper Access Control
CVE-2020-14500 Secomea GateManager Unauthenticated Remote Code Execution	9.8	• Impending Threat High-Value Target	Network pivot	Improper Access Control
CVE-2020-26085 Cisco Jabber Remote Code Execution Patch Bypass	9.9	• Impending Threat High-Value Target	Remote code execution	Injection
CVE-2020-3495 Cisco Jabber XHTML-IM XSS & Remote Code Execution	9.9	• Impending Threat High-Value Target	Remote code execution	Injection
CVE-2020-17008 splWOW64 Elevation of Privilege Patch Bypass	TBD	• Impending Threat Technical Details Widely Available	Local code execution	Memory Corruption

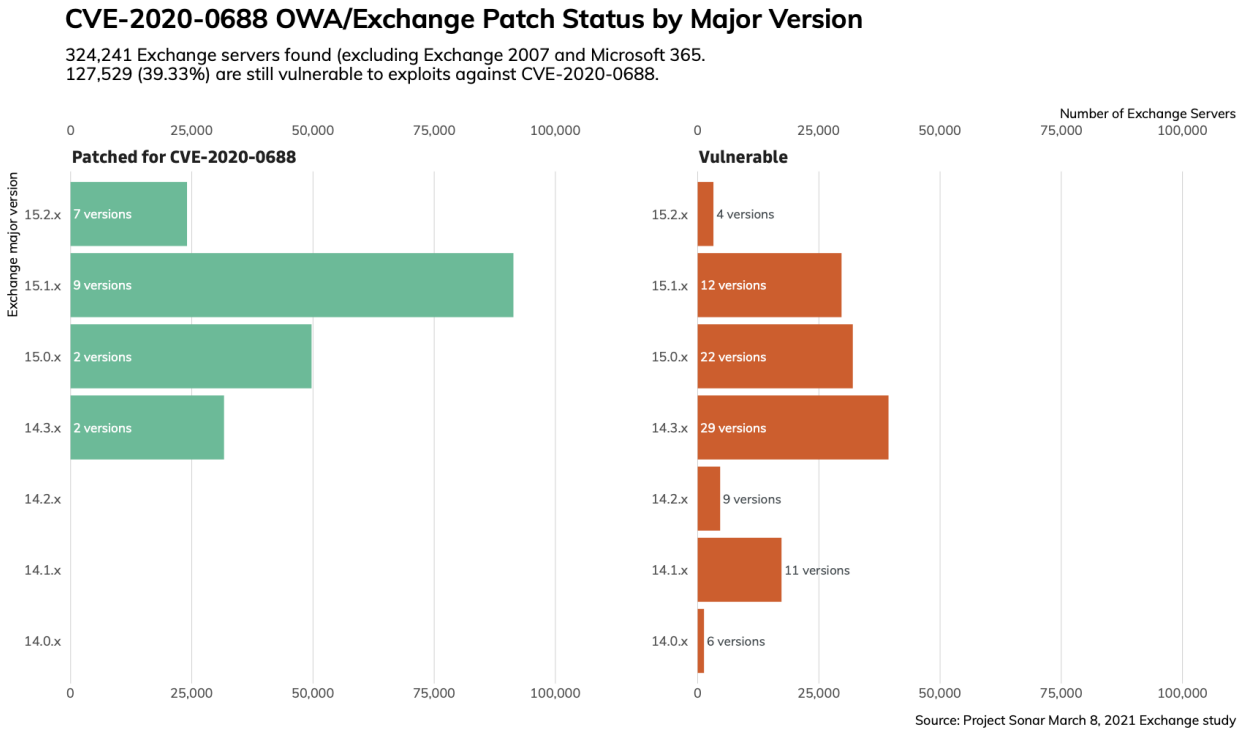
CVE-2020-0609 Windows Remote Desktop Gateway Remote Code Execution "BlueGate"	9.8	• Impending Threat Technical Details Widely Available	Remote code execution	Memory Corruption
CVE-2020-1170 Windows Defender Local Privilege Escalation	7.8	• Impending Threat Technical Details Widely Available	Local code execution	Improper Access Control
CVE-2020-1337 Windows Print Spooler Service Arbitrary File Write "PrintDemon" Patch Bypass #1	7.8	• Impending Threat Technical Details Widely Available	Local code execution	Improper Access Control
CVE-2020-6926 HP Device Manager Remote Method Invocation/Backdoor Database User	9.9	• Impending Threat Technical Details Widely Available	Network infrastructure compromise	Deserialization / Java
CVE-2020-17132 Microsoft Exchange Server DLP Policy Remote Code Execution Patch Bypass	8.4	• Impending Threat Technical Details Widely Available	Remote code execution	Injection

The “exploit available” references in this table all represent Metasploit modules that have been developed and tested for compatibility across a range of platforms. Metasploit is not the only toolkit we consider to be mature as far as capability goes—the Zerologon exploit built on Impacket, for instance, would have met the “exploit available” bar even before it was ported to Metasploit Framework. In the same vein, “high-value target” is a subjective term, though a common-sense one given the broad popularity of the products we’ve listed. “Technical details widely available” denotes in-depth, public information on identifying, triggering, and (frequently) exploiting the vulnerability in question. Technical details commonly include proof-of-concept code or demonstrations.

When a vulnerability occurs in a product with demonstrable value to attackers and a recent history of exploitation, it stands to reason that the intent component of the threat triangle is present even if there's not yet evidence of successful in-the-wild exploitation. There might be no better example of this principle than Microsoft Exchange and SharePoint vulnerabilities, three of which are notable impending threats (SharePoint CVE-2020-16952, and Exchange CVEs 2020-16875 and 2020-17132). Both products have remained consistently attractive targets for zero-day research, exploit development, and in-the-wild attacks over the years, which is no surprise given that vulnerabilities in Exchange and SharePoint environments can facilitate full compromises of Active Directory.

> "We consider most critical-rated and many important-rated CVEs in Microsoft Exchange and SharePoint to be strong impending threat candidates, regardless of whether exploit code has been released."

Unfortunately, patch rates continue to lag, as seen in the following data:



A few other CVEs in this group are worth highlighting for their value to would-be attackers: HP Device Manager CVE-2020-6926 and VMware vCenter Server CVE-2020-3952 are severe vulnerabilities in management software for Windows thin clients and virtualization infrastructure, respectively. Exploitation of these vulns gives an attacker broad access to not only the target software itself, but also all the downstream assets managed by that software. SaltStack Salt CVEs 2020-16846 and 2020-25592 are also severe vulnerabilities that, when chained, offer unauthenticated, remote attackers the opportunity to gain root access to the

target system and fully compromise networked infrastructure. Finally, lest we forget that Cisco's Jabber instant messaging platform is still in use, CVE-2020-3495 presents local attackers with half of a simple attack chain that can be used to worm corporate networks.

A New CVE Times Two (or Three): Patch Bypasses

2020 patch bypass CVEs:

CVE	CVSS v3	Threat Status	Attacker Utility	Vulnerability Class
CVE-2020-3950 VMware Fusion Local Privilege Escalation (Incomplete Patch)	7.8	• Impending Threat Exploit Available	Local code execution	Improper Access Control
CVE-2020-26085 Cisco Jabber Remote Code Execution Patch Bypass	9.9	• Impending Threat High-Value Target	Remote code execution	Injection
CVE-2020-1170 Windows Defender Local Privilege Escalation	7.8	• Impending Threat Technical Details Widely Available	Local code execution	Improper Access Control
CVE-2020-17008 splWOW64 Elevation of Privilege Patch Bypass	TBD	• Impending Threat Technical Details Widely Available	Local code execution	Memory Corruption
CVE-2020-1337 Windows Print Spooler Service Arbitrary File Write "PrintDemon" Patch Bypass #1	7.8	• Impending Threat Technical Details Widely Available	Local code execution	Improper Access Control
CVE-2020-17132 Microsoft Exchange Server DLP Policy Remote Code Execution Patch Bypass	8.4	• Impending Threat Technical Details Widely Available	Remote code execution	Injection

CVE-2020-3992 VMware ESXi OpenSLP Use-After-Free Remote Code Execution (Incomplete Patch)	9.8	• Targeted Threat Exploited in the wild	Network infrastructure compromise	Memory Corruption
CVE-2020-17496 vBulletin subWidgets data RCE	9.8	• Widespread Threat Exploited in the wild (released as 0 day)	Remote code execution	Injection
CVE-2020-14750 Oracle WebLogic Unauthenticated Remote Code Execution Patch Bypass	9.8	• Widespread Threat Exploited in the wild	Remote code execution	Improper Access Control

When security and IT teams disrupt scheduled patch cycles to remediate a new vulnerability or threat, they do so with the assumption that the vendor-supplied patch or mitigation has been developed with a critical eye and is (at least somewhat) trustworthy. Attackers work off a wholly different guiding principle—namely, that the discovery of one flaw means there are likely more just waiting in the wings, frequently in the same function, protocol, or section of the target product's code base. However, it hardly makes sense for attackers to put effort into hunting for zero-day vulnerabilities when they can change a single character in an exploit for a patched vulnerability and blow past the patch as if it weren't even there.

2020 revealed a number of patch bypass CVEs and incomplete fixes across a large cross-section of vendors, products, web applications, operating systems, and libraries. We detail nine of them here, but these are far from all-inclusive. 2020 was a banner year for patch bypasses, which is unfortunate for the many organizations whose patch cycles were disrupted or whose risk increased without their knowledge. Many of the patch bypasses we include in this report also have parent vulns that have either been actively exploited in the wild (e.g., Oracle WebLogic Server CVE-2020-14750, Windows kernel CVE-2020-17008, vBulletin CVE-2020-17496) or are widely documented (e.g., Windows "PrintDemon" CVE-2020-1337, which, incidentally, *also* has a patch bypass).

What might explain this vulnerability déjà vu? In some cases, a vulnerability's complexity can make solutions correspondingly complex to develop quickly and well. In fact,

> Many of the patch bypass techniques Rapid7 researchers evaluated over the course of the year were possible because filtering was added to stages of the original attack chain in lieu of addressing the bug's root cause.

Microsoft Exchange CVE-2020-16875 is a good example of this, where the “fix” mitigated the disclosed attack chain instead of the root cause, leading to the disclosure of patch bypass CVE-2020-17132.

It can also be difficult to address root cause when a vulnerability occurs in a component or behavior that’s closely related to the core architecture of the host system, which may not be able to be modified without adversely affecting legitimate use. Sometimes this occurs when the vulnerability is a feature that can be abused within the application, as with vBulletin CVE-2020-17496. The vulnerability in this case was related to how PHP templates were processed—intended and important functionality within vBulletin. If you’re looking for a second use case, this PyYAML library vulnerability is a good one. In both cases, the maintaining projects opted to implement filtering and other controls to restrict access to the features that had been abused. Ultimately, of course, those controls were bypassed.

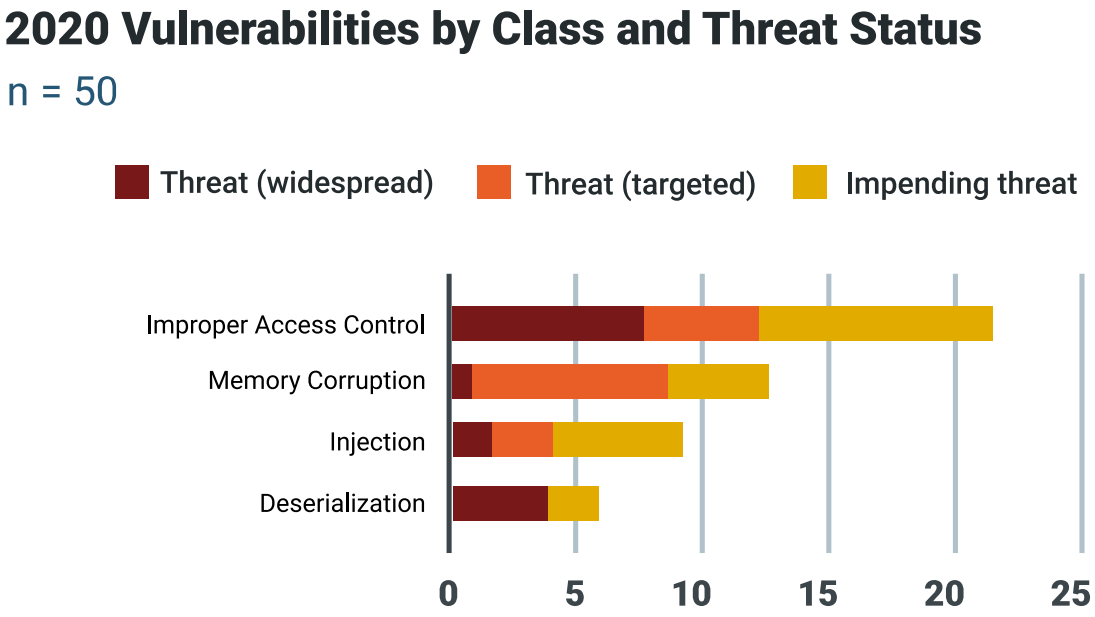
The prevalence of patch bypasses over the past year underscores the need for defense-in-depth strategies that include awareness of critical or exposed systems. See the guidance section at the end of this report for high-level recommendations.

The Dream of a Common Language

The Dream of a Common Language Practical Vulnerability Shorthand

When getting to know a new vulnerability, the first thing our research teams look for is an understanding of root cause and what an attacker might use the bug to achieve. Vulnerabilities arise from hundreds of conditions spanning all layers of the stack—from application programming errors to cryptographic implementations to hardware bugs and beyond. Likewise, the potential impact of any given vulnerability can vary widely based on implementation, security controls, and the sensitivity of the data or permissions an attacker can obtain as a result of exploitation. (To use an example from our own data, a remote file disclosure vulnerability might yield as little as benign source code files or as much as sensitive account credentials.)

As you may have noticed by now, we've characterized the vulnerabilities in our report dataset by their attacker utility and vulnerability class in addition to separating them by threat status. Attacker utility describes what an attacker can hope to gain as a result of successful exploitation—this often maps to a part of an exploit chain—while vulnerability class is an umbrella term that encompasses both root cause and the type of high-level technique that might trigger it.



We've defined four vulnerability classes—improper access control, memory corruption, injection, and deserialization—that are useful for making initial assessments of readily available attacker tooling and forming high-level hypotheses on relative exploitability. **Injection** attacks, for instance, use specially crafted input and techniques (e.g., SQL injection, operating system command injection) to compromise data integrity or run arbitrary code as a high-privileged user. These attacks tend to be stable and reliable, which makes them less likely to knock over systems than, say, attacks leveraging memory corruption flaws. **Deserialization** vulnerabilities come with a reputation for high exploitability and have a wealth of off-the-shelf tools with which to build exploit chains. And **improper access control** flaws—the vulnerability class most represented in both our dataset overall and in the widespread threats we've included—run the gamut from complex weaponization requirements to trivial exploitability using curl.

> Deserialization and improper access control bugs together are responsible for 11 of the most severe and impactful vulnerabilities of 2020.

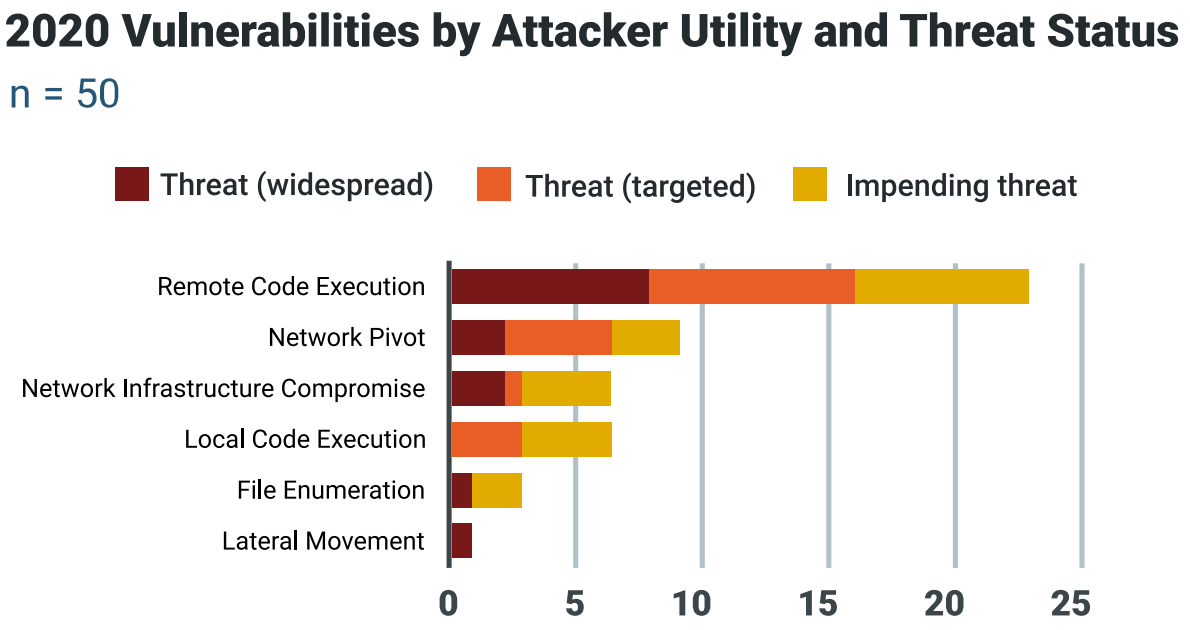
Improper access control vulnerabilities in particular posed a consistent threat to corporate networks, allowing attackers to bypass authentication (if it existed at all), execute code, add administrative users, and write files to disk. These flaws consistently blurred the line between vulnerability management and incident response and accounted for many of 2020's compromise investigations.

CVE	CVSS v3	Threat Status	Attacker Utility	Vulnerability Class

CVE-2019-18935 Telerik UI RadAsyncUpload .NET Deserialization	9.8	• Widespread Threat Exploited in the wild	Remote code execution	Deserialization / .NET
CVE-2020-0688 Microsoft Exchange Server Static Validation Key Remote Code Execution	8.8	• Widespread Threat Exploited in the wild	Remote code execution	Deserialization / .NET
CVE- 2020-10189 Zoho ManageEngine Remote Code Execution	9.8	• Widespread Threat Exploited in the wild (released as 0day)	Network infrastructure compromise	Deserialization / Java
CVE-2020-15505 MobileIron Core and Connector Remote Code Execution	9.8	• Widespread Threat Exploited in the wild	Remote code execution	Deserialization / Java
CVE-2020-1472 NetLogon Elevation of Privilege "Zerologon"	10	• Widespread Threat Exploited in the wild	Lateral movement	Improper Access Control
CVE-2020-11651 SaltStack Salt Remote Code Execution	9.8	• Widespread Threat Exploited in the wild	Network infrastructure compromise	Improper Access Control
CVE-2019-19781 Citrix NetScaler ADC/Gateway/SD-WAN Arbitrary Code Execution	9.8	• Widespread Threat Exploited in the wild	Network pivot	Improper Access Control
CVE-2020-5902 F5 Big-IP TMUI Remote Code Execution	10	• Widespread Threat Exploited in the wild	Network pivot	Improper Access Control

CVE-2020-14750 WebLogic Unauthenticated Remote Code Execution Patch Bypass	9.8	• Widespread Threat Exploited in the wild	Remote code execution	Improper Access Control
CVE-2020-6287 SAP NetWeaver AS Java "RECON"	10	• Widespread Threat Exploited in the wild	Remote code execution	Improper Access Control
CVE-2020-14882 Oracle WebLogic Server Remote Code Execution	9.8	• Widespread Threat Exploited in the wild	Remote code execution	Improper Access Control

The second type of metadata our researchers define when analyzing emergent threats answers the question, “As an attacker, what does this vulnerability get me?” Remote code execution is how vendors and CVE numbering authorities (CNAs) frequently describe high-impact vulnerabilities, but in some cases, that description downplays the ways that severe CVEs can be used to further compromise a network.



Unsurprisingly, vulnerabilities that provide attackers with remote code execution opportunities—i.e., the ability to execute a payload on a target system—are the most represented type of utility across the 50 CVEs we analyzed, at nearly 50% of the total dataset. Local code execution features much less prominently, which is also hardly surprising since it's much easier to launch internet-scale attacks from, well, the internet. We've already discussed the attractiveness of vulnerabilities that allow attackers to compromise entire swaths of network infrastructure (e.g., virtualization or automation infrastructure) instead of just a single target.

> It's much rarer for file enumeration CVEs to headline in security news or social media, but information leaks are important primitives that can open up paths to code execution and turn post-authentication CVEs into pre-authentication vulns.

One of the best examples of this is CVE-2019-11510, a remote, unauthenticated arbitrary file read vulnerability in Pulse Secure VPNs whose exploitation disclosed sensitive information like passwords and private keys. That information gave attackers what they needed to execute commands as root on vulnerable VPN servers. (GitLab CVE-2020-10977 provides another more recent example!)

All of this brings us to our final attacker utility type—one that played such a major role in 2020's threat landscape that it deserves a section all its own.

Hold the Door Open: Network Pivots

2020 network pivot CVEs:

CVE	CVSS v3	Threat Status	Attacker Utility	Vulnerability Class
CVE-2020-5135 SonicWall SonicOS Portal Buffer Overflow	9.4	• Impending Threat High-Value Target	Network pivot	Memory Corruption
CVE-2020-2021 Palo Alto Networks PAN-OS SAML Authentication Bypass and Remote Code Execution	10	• Impending Threat High-Value Target	Network pivot	Improper Access Control
CVE-2020-14500 Secomea GateManager Unauthenticated Remote Code Execution	9.8	• Impending Threat High-Value Target	Network pivot	Improper Access Control

CVE-2020-12271 Sophos XG Firewall Unauthenticated SQL Injection	9.8	• Targeted Threat Exploited in the wild (reported as 0day)	Network pivot	Injection
CVE-2020-8195 Citrix NetScaler ADC/Gateway/SD-WAN WANOP Improper Access Control	6.5	• Targeted Threat Exploited in the wild	Network pivot	Improper Access Control
CVE-2020-8193 Citrix NetScaler ADC/Gateway Authentication Bypass	6.5	• Targeted Threat Exploited in the wild	Network pivot	Improper Access Control
CVE-2020-8196 Citrix NetScaler ADC/Gateway/SD-WAN WANOP Improper Access Control	4.3	• Targeted Threat Exploited in the wild	Network pivot	Improper Access Control
CVE-2019-19781 Citrix NetScaler ADC/Gateway/SD-WAN Arbitrary Code Execution	9.8	• Widespread Threat Exploited in the wild	Network pivot	Improper Access Control
CVE-2020-5902 F5 Big-IP TMUI Remote Code Execution	10	• Widespread Threat Exploited in the wild	Network pivot	Improper Access Control

The ability to pivot from an external network to an internal network is a significant milestone for attackers. A **network pivot**—aptly if unimaginatively named—gives the attacker visibility into both internal and external traffic, most often by exploiting internet-facing systems such as VPNs, firewalls, routers, and other gateway devices. Network pivots aid in data exfiltration, traffic sniffing, and further attacks within the target network; they are, in short, extremely valuable to both state-sponsored and low-skilled attackers, along with penetration testers the world over.

Some of the past year's most high-profile and widely exploited vulnerabilities fell into the network pivot category. The first vulnerability to see large-scale exploitation in 2020 was Citrix NetScaler/Application Delivery Controller CVE-2019-19781, a directory traversal flaw that made its debut in December 2019 and harried incident responders well into the new

year. The vulnerability allowed remote, unauthenticated attackers to execute code on vulnerable gateways—and it wasn't the only Citrix CVE that acted as a network pivot point for attackers in 2020. The NSA warned in October that Chinese state-sponsored actors were exploiting three other vulnerabilities in Citrix's Application Delivery Controller, at least one of which could be used to extract valid VPN sessions with which to establish a foothold on internal networks.

For many network defenders, June 29 through July 29, 2020 was a particularly nightmarish stretch of an already challenging year: No fewer than four CVSS 10 vulnerabilities hit advisories, mailing lists, and news alerts during this period, three of which occurred within two weeks of one another. On June 29, Palo Alto Networks published a security advisory for CVE-2020-2021, a SAML authentication bypass affecting the company's PAN-OS operating system, which runs on gateways, portals, VPNs, and the firm's next-generation firewalls. SAML, or security assertion markup language, was (fortunately) not the default authentication scheme, and the company reinforced that attackers would need network access for successful exploitation. Still, an initial Rapid7 Labs study revealed just under 70,000 instances of Palo Alto Networks' Global Protect VPN on the public internet—an enticing pool of targets for attackers looking to access protected resources and find a way into corporate networks.

A day later, on June 30, 2020, F5 Networks quietly published a security bulletin advising customers of CVE-2020-5902, a severe remote code execution vulnerability in "undisclosed pages" of the Traffic Management User Interface (TMUI) of its BIG-IP product. Within a few days, multiple confirmations of exploitation in the wild had surfaced, along with reports of coin miners and automated secret scrapers targeting vulnerable BIG-IP instances, several thousand of which were exposed to the internet. F5 disclosed on July 8 that their previously released mitigation advice was able to be circumvented, and that anyone whose TMUI had been exposed to the internet should invoke incident response procedures—a series of events similar to Citrix CVE-2019-19781 earlier in the year, albeit on a bigger scale.

It wasn't until the end of July, 2020 that security firm Claroty released research detailing multiple critical vulnerabilities affecting various VPN implementations used primarily by industrial control systems (ICS) and operational technology (OT) networks. Taken together, the CVEs allowed attackers to decrypt VPN traffic, execute code remotely, and pilfer credentials for use in further attacks. If that weren't enough, it transpired that some 1,900 odd VPN gateways were exposing their Telnet administration port to the public internet.

In each of these cases, the gateway position of the vulnerable products amplified the vulnerabilities' severity and deepened the impact of exploitation. In October, several weeks after Zerologon exploitation had wrought havoc on vulnerable domain controllers, the U.S. Cybersecurity Infrastructure and Security Agency (CISA) published an alert warning that APT actors were chaining network pivot vulnerabilities with Zerologon in attacks against federal, state, local, and tribal networks, in addition to critical infrastructure and elections organizations. Several of the vulnerabilities CISA listed as potential targets—including Palo

Alto Networks, F5, and Citrix flaws discussed above—are included in this report as potential or active threats. They aren't alone: As a start, any of the vulnerabilities identified as a network pivot in our data could be used to similar ends. Their utility to attackers should make them a high priority for defenders.

Blues, Bleeds, Neighbors, and Ghosts: Memory Corruption Vulnerabilities

2020 memory corruption CVEs:

CVE	CVSS v3	Threat Status	Attacker Utility	Vulnerability Class
CVE-2020-5135 SonicWall SonicOS Portal Buffer Overflow	9.4	• Impending Threat High-Value Target	Network Pivot	Memory Corruption
CVE-2020-16898 Windows TCP/IP Remote Code Execution "Bad Neighbor"	8.8	• Impending Threat High-Value Target	Remote code execution	Memory Corruption
CVE-2020-17008 splWOW64 Elevation of Privilege Patch Bypass	TBD	• Impending Threat Technical Details Widely Available	Local code execution	Memory Corruption
CVE-2020-0609 Windows Remote Desktop Gateway Remote Code Execution "BlueGate"	9.8	• Impending Threat Technical Details Widely Available	Remote code execution	Memory Corruption
CVE-2020-17087 Windows Kernel Local Privilege Escalation	7.8	• Targeted Threat Exploited in the wild (reported as 0day)	Local code execution	Memory Corruption

CVE-2020-0986 Windows Sandbox Escape via splw64 Untrusted Pointer Dereference	7.8	• Targeted Threat Exploited in the wild (reported as 0day*)	Local code execution	Memory Corruption
CVE-2020-1020 Windows Adobe Font Manager Library Remote Code Execution	8.8	• Targeted Threat Exploited in the wild (reported as 0day)	Remote code execution	Memory Corruption
CVE-2020-14871 Oracle Solaris PAM Remote Code Execution	10	• Targeted Threat Exploited in the wild (reported as 0day)	Remote code execution	Memory Corruption
CVE-2020-15999 Google Chrome FreeType Heap Buffer Overflow	6.5	• Targeted Threat Exploited in the wild (reported as 0day)	Remote code execution	Memory Corruption
CVE-2020-3992 VMware ESXi OpenSLP Use-After-Free Remote Code Execution (Incomplete Patch)	9.8	• Targeted Threat Exploited in the wild	Network infrastructure compromise	Memory Corruption
CVE-2020-3118 Cisco IOS XR Software Discovery Protocol Format String Vulnerability "CDPwn"	8.8	• Targeted Threat Exploited in the wild	Remote code execution	Memory Corruption
CVE-2020-1350 Windows DNS Server Remote Code Execution "SigRed"	10	• Targeted Threat Exploited in the wild	Remote code execution	Memory Corruption

CVE-2020-0796 Windows SMBv3 Client/Server Remote Code Execution "SMBGhost"	10	• Widespread Threat Exploited in the wild	Remote code execution	Memory Corruption

*Kaspersky reported CVE-2020-0986 after they had seen in-the-wild attacks targeting the flaw, but said that Microsoft had already prepared a patch by the time they notified them. We still classify this CVE as a zero-day even if Kaspersky was not the only company to observe exploitation.

As many security practitioners who have experienced the transition into a post-WannaCry world can attest, MS17-010 and the ETERNALBLUE-based attacks of 2017 created a new paradigm for how the information security industry reacts to vulnerabilities in Windows network protocols like SMB and RDP (Remote Desktop Protocol). Patch Tuesday protocol CVEs instill dread in network security teams, and the infosec zeitgeist is quick to christen protocol vulnerabilities with a variety of “Blue”- and “Bleed”-related nicknames while security news keeps the term “wormable” fresh in our minds. None of it is unfair, unwarranted, or technically incorrect...and yet.

2020 security headlines trumpeted at least four wormable vulnerabilities in Windows networking protocols, as well as a smattering of others across major vendors. Despite the notoriety these bugs commanded, the prophesied worms failed to materialize, and with the exception of “SMBGhost” (as mentioned earlier, even this distinction is dubious), not one has seen exploitation at scale. Why?

Most any vulnerability can be exploited given sufficient skill, opportunity, and motivation. But certain bugs are much more difficult to trigger and weaponize reliably. We’re speaking, of course, of memory corruption vulnerabilities, the wild child of the vulnerability research world and the target of famous exploits like ETERNALBLUE and (to a much lesser extent) BlueKeep. Memory corruption is a large category of vulnerabilities that involves the misuse of data to alter memory and cause unexpected behavior. The overarching memory corruption category also contains more granular vulnerability types like stack and heap buffer overflows, type confusion, and use-after-free (UAFs).

When exploited successfully, memory corruption vulnerabilities can result in arbitrary code execution, or else in unhandled exceptions that cause an application to crash, triggering a denial of service (DoS) condition. The former is more frequently seen in local attacks, while the latter figures more prominently in remote attacks against internet-accessible targets. In recent years, broad adoption of safeguards has significantly complicated exploitation of memory corruption vulnerabilities. As a result, developing successful attacks takes skill and often requires manual secondary reconnaissance (e.g., finding a heap or kernel memory leak

to execute an exploit successfully), which limits utility at scale. Even when a memory corruption exploit is successful at scale—i.e., ETERNALBLUE—reliability is likely to remain a challenge.

On the other hand, exploits written for memory corruption vulns have a few compelling advantages over easier-to-develop attacks. Arbitrary code execution within a process or other memory space conceals malicious code better and leaves fewer artifacts (e.g., touching disk, spawning new processes), which translates to better operational security for malicious actors. These types of vulnerabilities are also likely to be present in targets straight out of the box, frequently in base-level functionality that doesn't require authentication or separate installation. Take October's "Bad Neighbor" vulnerability (CVE-2020-16898), for example: The CVE affected all versions of Windows by default without needing to account for configuration—uniform attack surface area even if exploitation is finickier.

> It's not surprising that "Bad Neighbor," "BlueGate," "SigRed," and several more of the year's high-profile CVEs never saw the release of highly effective public tools, or in many cases, any public tooling whatsoever. Turning memory corruption DoS actions into fully weaponized remote code execution is non-trivial.

But even in the absence of public tooling, it would be surprising—to the point of near impossibility—if fully weaponized exploits for these types of bugs weren't developed privately.

State-sponsored actors and other sophisticated attackers typically aren't resource-constrained the way in-house researchers at security firms or other corporate environments are. Neither, to a lesser degree, are vulnerability developers who sell zero-day exploits to the highest bidder—either directly or through brokers who act as a black box for both sides of the transaction. With that in mind, it's understandable that a number of 2020's memory corruption vulnerabilities saw exploitation by state-sponsored actors, whether they were disclosed as zero-days or included in intelligence agency communications about foreign adversary operations months after their release. Oracle Solaris CVE-2020-14871 and Windows kernel CVE-2020-0986 both fit the former description, while SigRed (CVE-2020-1350) and Cisco IOS XR CVE-2020-3118 fit the latter. The IOS XR vulnerability is a fantastic example of a bug that is entirely impractical at scale—successful exploitation requires an attacker to be on the local area network, directly attached to the vulnerable switch—but is evidently a bit more useful to state-sponsored threat actors.

We will see more of these types of bugs in 2021; we may even see more high-severity memory corruption bugs in Windows network protocols. While we share the security community's concern in many of these cases, "patch but don't panic" remains a good mantra to live by.

> Clicks and likes may be reasonable metrics for notoriety, but they're poor proxies for true risk.

Spotlight: Operational Technology

Spotlight: Operational Technology Securing Physical Systems in a

Digital World

The following section was written by the SCADAfence research team. SCADAfence is a global leader in securing operational technology (OT) and Internet of Things (IoT) networks for enterprises, industrial organizations, and critical infrastructure. Learn more here.

2020 witnessed a number of high-severity vulnerabilities that threatened OT and industrial networks and underlined the challenges of deeply embedded, often customized implementations of vulnerable technologies. But 2020's vulnerability suites, imbued with catchy names, aren't the first potential threats to highlight growing risk across an industry segment that's been caught in the middle of some of the world's largest cyberattacks.

First, some history: Operational technology (OT) systems are computer-operated systems that control physical and often critical processes. Many of these systems date back decades, and it was evident rather quickly that OT environments were not ready for the internet age. The initial response was to air-gap these systems by disconnecting all network connectivity. As it turns out, however, that approach was rather impractical. OT companies needed many of the benefits of connectivity to help them stay updated and competitive—benefits like improved monitoring and logistics, remote support, software updates, the ability to develop new connected product offerings, and so on.

OT asset owners, therefore, had to address the challenges of minimizing risk in an increasingly connected world. Legacy systems are still widely deployed in operational technology environments, the majority of which have not been designed with security in mind. It's an inescapable truth that large numbers of vulnerabilities exist in virtually every deployed OT system—from heavy equipment and industrial routers to internet-facing portals and common software libraries. These vulnerabilities affect millions of devices and hundreds of vendors, and are frequently difficult to mitigate or even identify across individual vendor implementations.

The Modern OT Threat Landscape

Threats to operational technology environments today include common and targeted ransomware along with targeted campaigns like Trisis or the 2015 attack on the Ukrainian power grid. In 2020, OT and IoT vendors also found themselves responding to a variety of vulnerabilities in common low-level software libraries—namely, common TCP/IP libraries shared by many vendors and products. These groups of CVEs, named Ripple20 and Amnesia:33, pose a challenge to both vendors tasked with developing fixes and OT organizations tasked with reducing risk. Vendor and supply chain fragmentation means that very often, millions of devices are affected by these CVEs, and the full list of vulnerable products and vendors may be populated over the course of months or years.

Whenever there are new vulnerabilities in shared libraries, the level of exploitability varies from device to device; as you might imagine, this creates quite a lot of confusion and adds another layer of complexity to the already complex industrial vulnerabilities space. Earlier this year, SCADAfence researchers tested one of the more severe Ripple20 vulnerabilities in our lab. CVE-2020-11898 has a CVSS score of 9.1 (critical) and can potentially allow remote attackers to read sensitive information. After testing on multiple devices that were reported vulnerable by their vendors, we found the vulnerability to be non-exploitable.

SCADAfence researchers also tested the exploitability of several of the more serious Amnesia:33 vulnerabilities and determined that exploitation was extremely unlikely outside of specific, tailor-made attack scenarios, since implementations vary from device to device. Furthermore, of the four remote code execution vulnerabilities released as part of the Amnesia:33 research, three of them reside in the way DNS domain names are encoded and the way DNS response packets are processed; another arises from the way IPv6 ICMP echo packets are processed. Why is this relevant? Because compared to other industries, it's *much* less common for OT environments to use DNS over hard-coded IP addresses, and most devices these days—especially OT networks—use IPv4 over IPv6.

There were, however, several vulnerabilities published in 2020 that do pose an active threat to OT environments and industrial organizations. Perhaps the most important set of 2020 vulnerabilities are CVE-2020-7486 and CVE-2020-7491, both of which affect Schneider Electric Triconex SIS devices. SIS (Safety Instrumented Systems) are critical process controls used to prevent disasters, like fires or explosions, in industrial environments. In July 2020, CISA in the U.S. warned that these vulnerabilities could be exploited remotely and with little skill. The same day, CISA and the NSA issued a joint alert recommending immediate actions to reduce exposure across OT and control systems.

OT Security Guidance

While asset owners often have comprehensive asset inventories of Windows endpoints, servers, and other devices, there is a large visibility gap when it comes to OT devices. This gap means that asset owners are not able to feed necessary information into their organizations' vulnerability management processes—information that includes comprehensive details on vendors, device models, versions, and network exposure. Managing risk is difficult if not impossible without an understanding of attack surface area introduced by OT devices.

Whenever critical vulnerabilities are published, and especially when they are known to be exploited, asset owners should perform assessments to determine the potential impact of the vulnerability and the organization's possible exposure. Some asset owners may also base their impact analyses solely on CVSS scores, but these can be misleading and lead to inefficient prioritization.

It helps when organizations are aware of common-sense facts when evaluating research or news articles on vulnerabilities affecting OT and IoT devices. Organizations that don't account for mitigating factors, such as uncommon configurations or unlikely attack scenarios, have a higher chance of investing resources in activities that don't end up reducing their risk.

Asset owners should also consider adopting automated tools that assist in the identification, prioritization, and remediation of OT vulnerabilities, in support of an OT vulnerability management program. Ideally, tooling should help OT network managers understand the size of their attack surface area and the probability of exploitation; this may include automated asset inventories, vulnerability correlation engines, and/or network exposure analyzers. For more information about this topic, please refer to the SCADAfence OT Vulnerability Management guide, which doubles as a comprehensive guide to industrial device patching.

Guidance for Defenders

Guidance for Defenders Key Takeaways From 2020 Threat

Responses

The tried-and-true pieces of guidance in this section have impeded exploitation for many of the vulnerabilities in this report and aided defenders in more quickly identifying signs of compromise or suspicious activity. We recognize that advice is rarely one-size-fits-all, and that security program maturity varies widely even among organizations of the same size, industry, and market cap. Nevertheless, the following suggestions encapsulate learnings that will apply to future threats as well as the specific risks highlighted in this paper:

> **Patch early and as often as possible, even when widespread exploitation isn't seen.**

When severe vulnerabilities are exploited within hours or days of publication, patching should be performed on an emergency basis, if possible. It is also wise to monitor for suspicious behavior and events in addition to fine-tuning intrusion detection and prevention systems. Relying on common rules and signatures alone is inadvisable during critical situations, as attackers routinely find ways to evade them. F5 BIG-IP CVE-2020-5902 is a fine example of this.

Several of the highest-severity vulnerabilities in this report weren't widely exploited until weeks or months after their publication (e.g., Zerologon, SMBghost, Microsoft Exchange Server CVE-2020-0688). Wherever possible, implementing a 30-day patch cycle will help protect against attacks with a longer tail, especially from commodity attackers who operate on economies of scale (e.g., botnet operators, ransomware campaigns, your everyday script kiddies, and so on).

> **Defense in depth is a more effective strategy than patching alone.**
>
> Skilled attackers are resourceful and, at times, utterly opportunistic. They can and will use any tool—any technique, any weakness, any piece of information—to build successful attack chains. Patches are not always effective, either, as evidenced by CVE-2020-14882 and CVE-2020-16875, which saw repeated patch bypasses.
>
> Additionally, many of the CVEs treated individually in this report can be used in concert with one or more additional vulnerabilities to achieve something beyond the scope of a single CVE's impact. Defenders can get ahead of future attacks by taking care not to treat individual vulnerabilities as if they existed in a vacuum, but instead choosing to implement controls and detection mechanisms across the whole of their environment.

Finally, ensuring that (preferably aggregated) logging is set up across networks and hosts will save some time during active threat events. There are several community-driven signature repositories and low-cost (or free!) rulesets that can give defenders at least basic visibility

into potential intrusions in their environments, along with a plethora of commercial solutions. Knowing ahead of time what kind of visibility defenders have into suspicious events will drive faster and more effective response during critical situations.

> **Keep an up-to-date inventory list that emphasizes assets or products that sit on the perimeter and/or may be used as pivot points for external attackers to gain access to the internal network.**

Understanding attack surface area and critical network entry points saves time when severe vulnerabilities surface in internet-facing technologies. Pay particular attention to security gateway products such as VPNs and firewalls, as well as anything else that's exposed by common practice or necessity. CVE-2019-19781 and CVE-2020-0688 are noteworthy examples.

Management and administrative interfaces should never be exposed to the public internet. The same goes for domain controllers and any other assets that organizations would not want an external attacker to be able to probe, such as IoT devices unwittingly exposed online. Audit internet-exposed attack surface area regularly, including via external penetration tests, if possible.

At Rapid7, we believe that research-driven context on vulnerabilities and emergent threats is critical to building forward-looking security programs and advancing community knowledge. Security and IT teams face mounting challenges in a heightened threat climate, and we are committed to partnering with those teams to foster deeper understanding of defense-in-depth strategies that will strengthen organizations' security posture both now and in the future.

For more information on the vulnerabilities featured in this report, and for Rapid7 and community analysis of new vulnerabilities and threats, keep an eye on AttackerKB. If you wish to follow updates on specific CVEs or add assessments of your own, you can create an account using GitHub.

Appendix

Why These CVEs?

Context is king when analyzing vulnerabilities, just as it is when managing them. Our investigations into root cause, exploitability, and potential attack vectors often start the same way as conversations among security stakeholders—with a vendor advisory that pops a red flag, a bit of private intel from a friend, a tweet thread, or a news article. To put it simply, the vulnerabilities in this report piqued our interest, as much for the community's reaction to them as for any bona fide threats that emerged. Notably, they also affect software and systems relied upon by many, from enterprises to end users. This dataset does not include all CVEs or even all active threats we evaluated in 2020, but it does represent a diverse sample of attacker use cases and exploitation case studies.

Of course, our intent is not to imply that any one CVE or vulnerability group is less important than others. Security teams, network administrators, and defenders at large have in-depth understanding of which assets are critical in their environments and how action taken may affect their business priorities. What we offer is an attacker-centric view of the vulnerability landscape that Rapid7 customers and the security community can use to inform the policies and practices that they employ as part of a larger defense-in-depth strategy.

Notes on Methodology

CVEs featured in this report are from 2020 with two exceptions: Telerik UI CVE-2019-18935 and Citrix NetScaler ADC/Gateway CVE-2019-19781, both of which were published late in 2019 and saw sustained exploitation throughout 2020.

Since the trustworthiness of our data is important, we cite primary sources wherever possible for vulnerabilities we've listed as exploited in the wild—that is, we reference firsthand accounts of exploitation from the organizations or individuals who detected, verified, and reported them. Examples of primary sources referenced throughout this paper include U.S. intelligence agency alerts of advanced persistent threat (APT) exploitation; security firm analyses of threats and IOCs they've tracked during incident response or other investigations; and vendor advisories that specify exploitation in the wild (this includes CVEs that are disclosed as zero-days). In the interest of readability, when firsthand reports of exploitation are disparate and varied, we will occasionally cite an article in a security news publication that aggregates those accounts.

The CVEs we have categorized as exploited in the wild in this report are not the only vulnerabilities actively exploited during the 2020 calendar year. For example, we have excluded a number of other exploited bugs in Internet Explorer, Chrome, and Firefox. Google Project Zero has a spreadsheet of some other zero-days exploited in the wild in 2020 here.

Glossary of Terms

Attacker Utilities

Remote code execution (RCE): Code execution on a remote target. Typically refers to the ability to execute a payload on a target system (e.g., obtain a shell session). Aids in credential stealing, data exfiltration, and so on.

Local code execution: The ability to run code locally on a system to which the attacker already has some access. Most commonly used to escalate privileges (e.g., by executing code as the user running the vulnerable application).

Network infrastructure compromise: Compromise of networked infrastructure, such as a network management system or backup system, that may give an attacker access to everything managed by that software. Vulnerabilities in virtualization, automation, and/or device management infrastructure all fall into this category.

Network pivot: The ability to pivot from an external network to an internal network, most often by exploiting internet-facing systems such as VPNs, firewalls, routers, and other gateway devices. A network pivot gives an attacker visibility into both internal and external traffic and aids in data exfiltration, traffic sniffing, and further attacks within the target network.

File enumeration: The ability to enumerate files on a target. File reads do not give an attacker a path to code execution by themselves, but instead function as primitives that allow attackers to gather information that enables a secondary part of an exploit chain (e.g., remote code execution). Can aid in turning a post-authentication vulnerability into a pre-authentication vulnerability.

Vulnerability Classes

Deserialization is the process through which an application is able to convert data from a portable format to data types native to its own language. Many modern languages support deserialization, including Java, .NET, Python, and Ruby. The deserialization process can pose a threat to security when the data that is loaded into the native language can be tampered with by a malicious party. Typical attacks involve configuring the data to invoke a method with the arguments necessary to execute an operating system command. This results in command execution in the context of the loading application. Common solutions to this security problem include cryptographically signing the data to ensure its authenticity and utilizing an allowlist of data types that are permitted to be loaded. Associated CWEs: CWE-502.

Improper Access Control refers to a missing or insufficient access control to a particular interface into a system (most often a remotely accessible API). Improper uses of cryptography for the purpose of authentication also fall under this vulnerability class. Common solutions to this problem include proper authentication, authorization, and accounting implementations for all sensitive interfaces, as well as secure management of all related secrets. A non-exhaustive list of associated CWEs: CWE-285, CWE-200, CWE-287, CWE-732.

Memory Corruption is a large category of vulnerabilities that involve the misuse of data through a variety of means to alter memory and produce unexpected behavior. This vulnerability class includes improper boundary enforcement, type confusion, uninitialized data use, and the use of data after it has been freed, to name a few. These vulnerabilities often manifest themselves in languages that are not considered "type-safe.” Successful exploitation of memory corruption vulnerabilities can result in arbitrary code execution within

the context of the running application, or in an unhandled exception that causes the application to crash and triggers a denial of service (DoS) condition. Common solutions to this problem typically involve additional validation on parameters to key operations, such as those used to load and store data. Successful exploitation of these classes of vulnerabilities has become more complex in recent years due to the variety of countermeasures and safeguards that have been developed, such as kASLR, Control Flow Guard, win32k Type Isolation, and so on. A non-exhaustive list of associated CWEs: CWE-787, CWE-125, CWE-416, CWE-190, CWE-476.

Injection is a large category of vulnerabilities involving specially crafted input that is interpreted in a particular way by an associated system. Most commonly seen in web applications, injection attacks are often more specifically labeled by the type of data being interpreted (e.g., SQL, LDAP, OS commands). The root cause of these vulnerabilities is almost always insufficient sanitization on data received from a malicious party. Exploitation of these vulnerabilities tends to be reliable, rarely resulting in service degradation unless intended (such as through SQL or OS commands).

The context under which the logic is executed typically depends on how it is interpreted. In the case of a web application, for example, SQL injection may be executed on a back-end database server, while OS commands are injected on the front-end web server, and JavaScript is executed by the end user's browser. This class of vulnerabilities is therefore unique in that it commonly involves a vulnerability in one system compromising the integrity of others. Common solutions to this problem typically involve implementing strict sanitization on parameters though the use of allowlists. A non-exhaustive list of associated CWEs: CWE-79, CWE-20, CWE-89, CWE-94.

Full Dataset

CVE	CVSS v3	Threat Status	Attacker Utility	Vulnerability Class
CVE-2020-3950 VMware Fusion Local Privilege Escalation (Incomplete Patch)	7.8	• Impending Threat Exploit Available	Local code execution	Improper Access Control
CVE-2020-25592 SaltStack Salt Authentication Bypass	9.8	• Impending Threat Exploit Available	Network infrastructure compromise	Improper Access Control

CVE-2020-3952 VMware vCenter Server/vmdir Information Disclosure	9.8	• Impending Threat Exploit Available	Network infrastructure compromise	Improper Access Control
CVE-2020-16846 SaltStack Salt Command Injection	9.8	• Impending Threat Exploit Available	Network infrastructure compromise	Injection
CVE-2020-16952 Microsoft SharePoint Authenticated Remote Code Execution	7.8	• Impending Threat Exploit Available	Remote code execution	Deserialization / .NET
CVE-2020-16875 Microsoft Exchange Server DLP Policy Remote Code Execution	8.4	• Impending Threat Exploit Available	Remote code execution	Injection
CVE-2020-8209 Citrix XenMobile Server File Disclosure	7.5	• Impending Threat High-Value Target	File enumeration	Improper Access Control
CVE-2020-3187 Cisco Adaptive Security Appliance/Firepower Threat Defense Path Traversal	9.1	• Impending Threat High-Value Target	File enumeration	Improper Access Control
CVE-2020-5135 SonicWall SonicOS Portal Buffer Overflow	9.4	• Impending Threat High-Value Target	Network pivot	Memory Corruption
CVE-2020-2021 Palo Alto Networks PAN-OS SAML Authentication Bypass and Remote Code Execution	10	• Impending Threat High-Value Target	Network pivot	Improper Access Control

CVE-2020-14500 Secomea GateManager Unauthenticated Remote Code Execution	9.8	• Impending Threat High-Value Target	Network pivot	Improper Access Control
CVE-2020-16898 Windows TCP/IP Remote Code Execution "Bad Neighbor"	8.8	• Impending Threat High-Value Target	Remote code execution	Memory Corruption
CVE-2020-26085 Cisco Jabber Remote Code Execution Patch Bypass	9.9	• Impending Threat High-Value Target	Remote code execution	Injection
CVE-2020-3495 Cisco Jabber XHTML-IM XSS & Remote Code Execution	9.9	• Impending Threat High-Value Target	Remote code execution	Injection
CVE-2020-17008 splWOW64 Elevation of Privilege Patch Bypass	TBD	• Impending Threat Technical Details Widely Available	Local code execution	Memory Corruption
CVE-2020-1170 Windows Defender Local Privilege Escalation	7.8	• Impending Threat Technical Details Widely Available	Local code execution	Improper Access Control
CVE-2020-1337 Windows Print Spooler Service Arbitrary File Write "PrintDemon" Patch Bypass #1	7.8	• Impending Threat Technical Details Widely Available	Local code execution	Improper Access Control
CVE-2020-6926 HP Device Manager Remote Method Invocation/Backdoor Database User	9.9	• Impending Threat Technical Details Widely Available	Network infrastructure compromise	Deserialization / Java

CVE-2020-0609 Windows Remote Desktop Gateway Remote Code Execution "BlueGate"	9.8	• Impending Threat Technical Details Widely Available	Remote code execution	Memory Corruption
CVE-2020-17132 Microsoft Exchange Server DLP Policy Remote Code Execution Patch Bypass	8.4	• Impending Threat Technical Details Widely Available	Remote code execution	Injection
CVE-2020-17087 Windows Kernel Local Privilege Escalation	7.8	• Targeted Threat Exploited in the wild (reported as 0day)	Local code execution	Memory Corruption
CVE-2020-10148 SolarWinds Orion API Authentication Bypass	9.8	• Targeted Threat Exploited in the wild (reported as 0day)	Network infrastructure compromise	Improper Access Control
CVE-2020-12271 Sophos XG Firewall Unauthenticated SQL Injection	9.8	• Targeted Threat Exploited in the wild (reported as 0day)	Network pivot	Injection
CVE-2020-1020 Windows Adobe Font Manager Library Remote Code Execution	8.8	• Targeted Threat Exploited in the wild (reported as 0day)	Remote code execution	Memory Corruption
CVE-2020-4006 VMware Workspace ONE Command Injection	9.1	• Targeted Threat Exploited in the wild (reported as 0day)	Remote code execution	Injection

CVE-2020-14871 Oracle Solaris PAM Remote Code Execution	10	• Targeted Threat Exploited in the wild (reported as 0day)	Remote code execution	Memory Corruption
CVE-2020-15999 Google Chrome FreeType Heap Buffer Overflow	6.5	• Targeted Threat Exploited in the wild (reported as 0day)	Remote code execution	Memory Corruption
CVE-2020-0986 Windows Sandbox Escape via splw64 Untrusted Pointer Dereference	7.8	• Targeted Threat Exploited in the wild (reported as 0day*)	Local code execution	Memory Corruption
CVE-2020-1048 Windows Print Spooler Service Arbitrary File Write "PrintDemon"	7.8	• Targeted Threat Exploited in the wild	Local code execution	Improper Access Control
CVE-2020-3992 VMware ESXi OpenSLP Use-After-Free Remote Code Execution (Incomplete Patch)	9.8	• Targeted Threat Exploited in the wild	Network infrastructure compromise	Memory Corruption
CVE-2020-8195 Citrix NetScaler ADC/Gateway/SD-WAN WANOP Improper Access Control	6.5	• Targeted Threat Exploited in the wild	Network pivot	Improper Access Control
CVE-2020-8193 Citrix NetScaler ADC/Gateway Authentication Bypass	6.5	• Targeted Threat Exploited in the wild	Network pivot	Improper Access Control

CVE-2020-8196 Citrix NetScaler ADC/Gateway/SD-WAN WANOP Improper Access Control	4.3	• Targeted Threat Exploited in the wild	Network pivot	Improper Access Control
CVE-2020-0601 Windows CryptoAPI Spoofing Vulnerability "Curveball"	8.1	• Targeted Threat Exploited in the wild	Remote code execution	Improper Access Control
CVE-2020-3118 Cisco IOS XR Software Discovery Protocol Format String Vulnerability "CDPwn"	8.8	• Targeted Threat Exploited in the wild	Remote code execution	Memory Corruption
CVE-2020-1350 Windows DNS Server Remote Code Execution "SigRed"	10	• Targeted Threat Exploited in the wild	Remote code execution	Memory Corruption
CVE-2020-10189 Zoho ManageEngine Remote Code Execution	9.8	• Widespread Threat Exploited in the wild (released as 0day)	Network infrastructure compromise	Deserialization / Java
CVE-2020-17496 vBulletin subWidgets data RCE	9.8	• Widespread Threat Exploited in the wild (released as 0day)	Remote code execution	Injection
CVE-2020-14750 WebLogic Unauthenticated Remote Code Execution Patch Bypass	9.8	• Widespread Threat Exploited in the wild (reported as 0day)	Remote code execution	Improper Access Control

CVE-2020-3452 Cisco ASA/FTD Web Services Read-Only Path Traversal Vulnerability	7.5	• Widespread Threat Exploited in the wild	File enumeration	Injection
CVE-2020-1472 NetLogon Elevation of Privilege "Zerologon"	10	• Widespread Threat Exploited in the wild	Lateral movement	Improper Access Control
CVE-2020-11651 SaltStack Salt Remote Code Execution	9.8	• Widespread Threat Exploited in the wild	Network infrastructure compromise	Improper Access Control
CVE-2019-19781 Citrix NetScaler ADC/Gateway/SD-WAN Arbitrary Code Execution	9.8	• Widespread Threat Exploited in the wild	Network pivot	Improper Access Control
CVE-2020-5902 F5 Big-IP TMUI Remote Code Execution	10	• Widespread Threat Exploited in the wild	Network pivot	Improper Access Control
CVE-2019-18935 Telerik UI RadAsyncUpload .NET Deserialization	9.8	• Widespread Threat Exploited in the wild	Remote code execution	Deserialization / .NET
CVE-2020-0688 Microsoft Exchange Server Static Validation Key Remote Code Execution	8.8	• Widespread Threat Exploited in the wild	Remote code execution	Deserialization / .NET
CVE-2020-0796 Windows SMBv3 Client/Server Remote Code Execution "SMBghost"	10	• Widespread Threat Exploited in the wild	Remote code execution	Memory Corruption

CVE-2020-14882 Oracle WebLogic Server Remote Code Execution	9.8	• Widespread Threat Exploited in the wild	Remote code execution	Improper Access Control
CVE-2020-15505 MobileIron Core and Connector Remote Code Execution	9.8	• Widespread Threat Exploited in the wild	Remote code execution	Deserialization / Java
CVE-2020-6287 SAP NetWeaver AS Java "RECON"	10	• Widespread Threat Exploited in the wild	Remote code execution	Improper Access Control

*Kaspersky reported CVE-2020-0986 after they had seen in-the-wild attacks targeting the flaw, but said that Microsoft had already prepared a patch by the time they notified them. We still classify this CVE as a zero-day even if Kaspersky was not the only company to observe exploitation.

References

Security research is a community pursuit. This report benefited from the work of many individual researchers and research teams, including but not limited to the work of the folks listed below:

Steven Seeley, Vulcan 360 (2020)

JJ Lehmann and Ofri Ziv, Guardicore (2020)

Rajesh Nataraj, Sophos (2020)

Kevin Beaumont, Microsoft (2020)

Nahuel Sanchez, Pablo Artuso, Onapsis (2020)

Tom Tervoort and Ralph Moonen, Secura (2020)

Research and Intelligence Fusion Team (RIFT), NCC Group (2020)

Sophos Labs (2020)

Dirk-jan (dirkjanm) (2020)

x41sec (X41 D-SEC GmbH) (2020)

Tom Sellers, Rapid7 Labs (2020)

Maddie Stone, Google Project Zero (2020)

James Forshaw, Google Project Zero (2020)

Johannes B. Ullrich, Ph.D, SANS Technology Institute (2020)

Aaron Soto, Rapid7 (2019)

Jon Hart, Rapid7 Labs (2019)

Grant Willcox, Rapid7 (2020)

Joe Needleman, Andrew Nelson, Tony Lee, and Mark Stevens, BlackBerry (2020)

Troy Mursch, Bad Packets (2019)

Shelby Pace, Rapid7 (2020)

Claroty Research Team, Claroty (2020)

Bob Rudis, Rapid7 Labs (2020)

John Miller, Matt Allen, Christopher Glyer, Ian Ahl, Nick Carr, FireEye (2017)

ollypwn (2020)

ZecOps Research Team, ZecOps (2020)

RiskSense (2018)

Brent Cook, Rapid7 (2019)

Craig Young, Tripwire (2020)

Daniel dos Santos, Stanislav Dashevskyi, Jos Wetzels, and Amine Amri, Forescout Research Labs (2020)

Maayan Fishelov, SCADAfence (2020)

Haozhe Zhang, Qi Deng, Zhibin Zhang and Ruchna Nigam, Palo Alto (2020)

Christopher Glyer, Dan Perez, Sarah Jones, Steve Miller, FireEye (2020)

Volexity Threat Research, Volexity (2020)

Tony Lambert, Red Canary Intel (2020)

Boris Larin, Kaspersky (2020)

Andy Reactor, Mandiant (2020)

Kevin Beaumont (2020)

SophosLabs (2020)
Mateusz Jurczyk, Google Project Zero (2020)

Justin Moore, Wojciech Ledzion, Luis Rocha, Adrian Pisarczyk, Daniel Caban, Sara Rincon, Daniel Susin, Antonio Monaca, FireEye (2020)

Ben Hawkes, Google Project Zero (2020)

Nick Bloor (2020)
Voidsec (2020)

Itm4n (2020)
Marcus Hutchins, Kryptos Logic (2020)

About Rapid7

Rapid7 (Nasdaq: RPD) is advancing security with visibility, analytics, and automation delivered through our Insight cloud. Our solutions simplify the complex, allowing security teams to work more effectively with IT and development to reduce vulnerabilities, monitor for malicious behavior, investigate and shut down attacks, and automate routine tasks. Customers around the globe rely on Rapid7 technology, services, and research to improve security outcomes and securely advance their organizations. For more information, or to get involved in our threat research, visit www.rapid7.com.