I Solemnly Swear My Driver Is Up to No Good: Hunting for Attestation Signed Malware

mandiant.com/resources/blog/hunting-attestation-signed-malware

During a recent Incident Response investigation, Mandiant discovered a malicious driver used to terminate select processes on Windows systems. In this case, the driver was used in an attempt to terminate the Endpoint Detection and Response (EDR) agent on the endpoint. Mandiant tracks the malicious driver and its loader as POORTRY and STONESTOP respectively. Soon after the initial discovery, Mandiant observed a POORTRY driver sample signed with a Microsoft Windows Hardware Compatibility Authenticode signature. Careful analysis of the driver's Authenticode metadata led to a larger investigation into malicious drivers signed via the Windows Hardware Compatibility Program. The investigation found a wider issue:

- The malicious drivers are signed directly by Microsoft and identifying the original software vendor requires inspecting the signature with code
- Several distinct malware families, associated with distinct threat actors, have been signed with this process
- Mandiant identified at least nine unique organization names associated with attestation signed malware

This research is being released alongside a blog post by our colleagues at SentinelOne.

Code Signing and the Windows Hardware Compatibility Program

Relationships are built on trust. The same goes for the relationship we have with the software we rely on when using our computers every day; do I trust the execution of this program, and why? Software can be very opaque to end users; when it claims to be from Company X, what mechanisms exist to verify software's trustworthiness?

[Queue John Cena walk-out music.]

Code signing has entered the ring.

Code signing is a means to ensure integrity and authenticity of a given file. Software vendors obtain certificates used for code signing from trusted Certificate Authorities (CA), who abide by standards set forth by the CA/Browser Forum and CA Security Council. These guidelines detail requirements, which include verifying the legal existence and identity of the company, and that the requestor of the certificate is authorized to act on behalf of the software vendor they claim to represent.

This certificate is then used to sign the software and provide a level of trust between the software and the operating system. Code signing enforcement policies differ per operating system and file type, from only allowing signed code to execute, to minimizing security warnings for execution of signed code, to purely serving as a digital signature denoting the authenticity of an application.

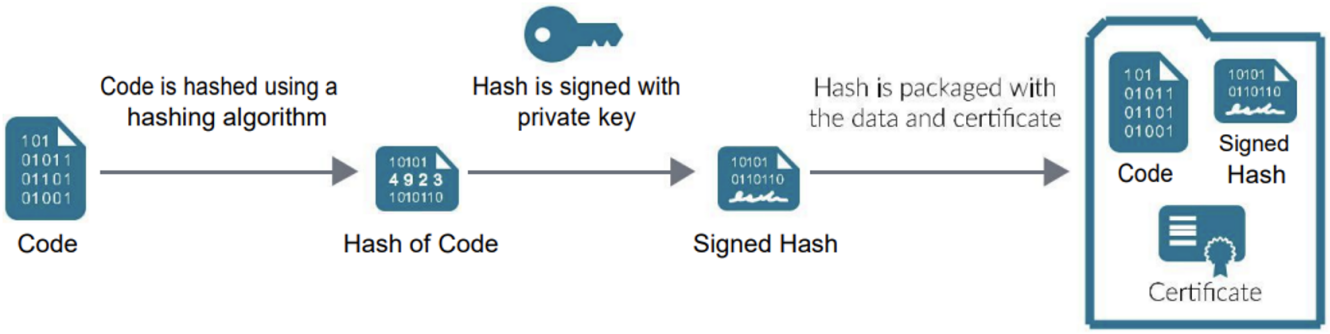

Figure 1: Code signing overview (source)

Microsoft's code signing implementation for Windows binaries is known as Authenticode. Authenticode has several features specific to drivers and driver packages, and assists hardware vendors in getting their drivers signed properly via the Windows Hardware Compatibility Program.

"The Windows Hardware Compatibility Program is designed to help your company deliver systems, software and hardware products that are compatible with Windows and run reliably on Windows 10, Windows 11 and Windows Server 2022. The program also provides guidance for developing, testing and distributing drivers. Using the Windows Hardware Dev Center dashboard, you can manage submissions, track the performance of your device or app, review telemetry and much more."

There are multiple phases to work through the Windows Hardware Compatibility Program process.

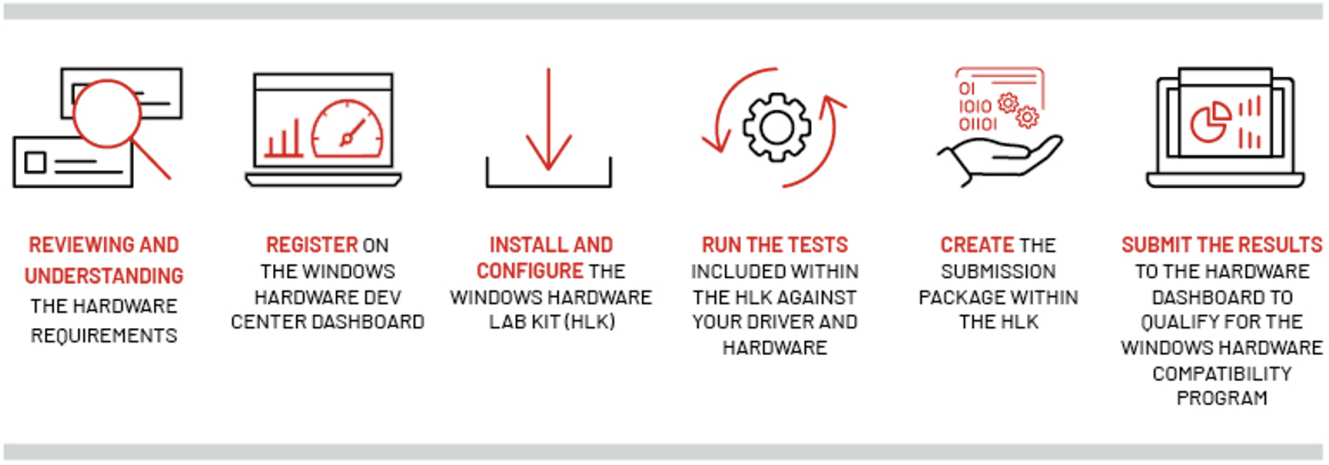

Figure 2: Steps in Windows Hardware Compatibility Program

For operability on Windows 10 and later, drivers can be submitted to Microsoft for **attestation signing**.

In this attestation signing process, digital signatures are used to verify the integrity of submitted driver packages and to verify the identity of the software publisher who provided the driver packages. This process requires that the submitting organization sign their driver package with an Extended Validation (EV) certificate, which has enhanced identification requirements over other code-signing certificates and must use stronger encryption algorithms. These EV certificates are offered by a smaller circle of Certificate Authorities who have agreed to enhanced auditing requirements.

As an additional step, vendors can submit their driver for Hardware Lab Kit (HLK) testing, to become Windows Certified. When a driver receives attestation signing, it's not Windows Certified. An attestation signature from Microsoft indicates that the driver can be trusted by Windows, but because the driver has not been tested in HLK Studio, there are no assurances made around compatibility, functionality, and so on.

At a high level, there are 9 steps to submit an attestation signed driver within the compatibility program process.

1. Register for the Hardware Developer program
2. Identify or purchase an Extended Validation (EV) certificate
3. Download and install the Windows Driver Kit (WDK)
4. Create the CAB file that will be submitted for approval. The CAB file includes the driver itself, driver INF, symbol file, and catalog files.
5. Sign the CAB file with the EV certificate
6. Submit the EV signed CAB via the hardware dashboard
7. Microsoft will sign the driver
8. Download signed driver from the hardware dashboard

9. Validate and test the signed driver

The output of this process is an attestation signed driver.

Mandiant has continually observed threat actors use compromised, stolen, and illicitly purchased code-signing certificates to sign malware, lending legitimacy and subverting security controls such as application allow-listing policies. Attestation signed drivers take the trust granted to them by the CA and transfers it to a file whose Authenticode signature originates from Microsoft itself. We assess with high confidence that threat actors have subverted this process using illicitly obtained EV code signing certificates to submit driver packages via the attestation signing process, and in effect have their malware signed by Microsoft directly.

Threat Data and Observations

Mandiant has observed UNC3944 utilizing malware that has been signed via the attestation signing process. UNC3944 is a financially motivated threat group that has been active since at least May 2022 and commonly gains initial network access using stolen credentials obtained from SMS phishing operations. In some cases, the group's post-compromise objectives have focused on accessing credentials or systems used to enable SIM swapping attacks, likely in support of secondary criminal operations occurring outside of victim environments.

UNC3944 has been observed deploying both STONESTOP and POORTRY as early as August 2022.

STONESTOP is a Windows userland utility that attempts to terminate processes by creating and loading a malicious driver. Mandiant tracks this malicious driver as POORTRY. POORTRY is a Windows driver that implements process termination and requires a userland utility to initiate the functionality. At driver entry it registers device `\device\KApcHelper1` for interaction by user-space utilities like STONESTOP.

Mandiant has observed signed POORTRY drivers dating back to June of 2022 with a mix of certificates, including stolen certificates that have been widely circulated. Usage of POORTRY appears across different threat groups and is consistent with malware available for purchase or shared freely between different groups.

Table 1: Additional signed POORTRY samples

Compile time	Signing time	MD5	Certificate Subject Common Name
2022-06-02 10:09:08	20220811 13:27:00	10f3679384a03cb487bda9621ceb5f90	Zhuhai liancheng Technology Co., Ltd.
2022-06-02 10:09:08		04a88f5974caa621cee18f34300fc08a	Zhuhai liancheng Technology Co., Ltd.
2022-06-02 10:09:08	20220915 15:49:00	6fcf56f6ca3210ec397e55f727353c4a	Microsoft Windows Hardware Compatibility Publisher
2022-06-06 15:14:46		0f16a43f7989034641fd2de3eb268bf1	NVIDIA Corporation
2022-08-20 15:19:01	20220821 05:43:00	ee6b1a79cb6641aa44c762ee90786fe0	Microsoft Windows Hardware Compatibility Publisher
2022-10-02 19:48:02	20221019 17:15:00	909f3fc221acbe999483c87d9ead024a	Microsoft Windows Hardware Compatibility Publisher

Unlike the earlier examples, many of which were improperly signed, this POORTRY sample is legitimately signed and verified with a Microsoft Windows Hardware Compatibility Publisher certificate. This is a Microsoft certificate that is used across the attestation program, and therefore is used extensively on legitimate binaries as well.

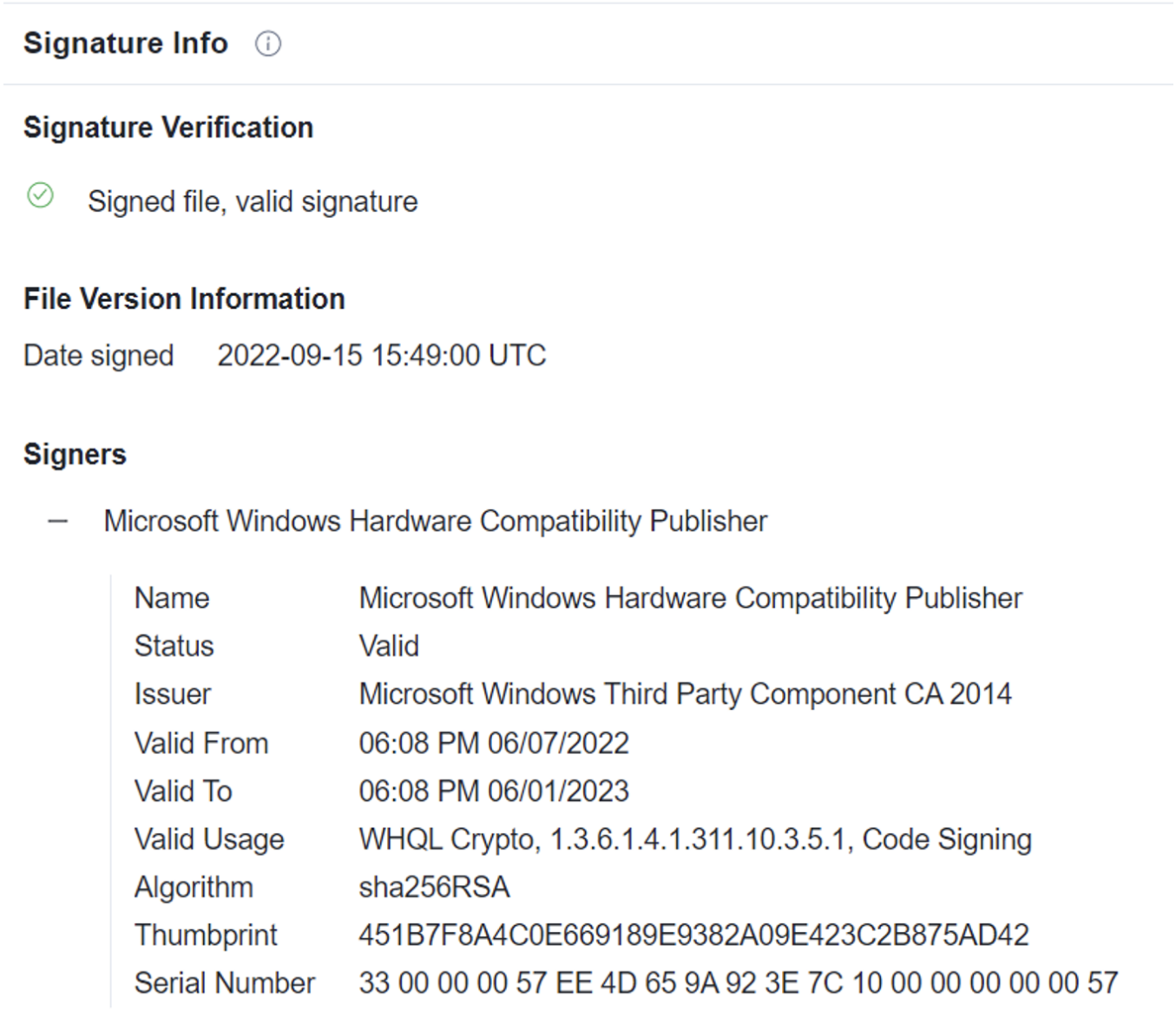

Figure 3: Valid POORTRY signature data

The public key used for the attestation signing (Appendix C: POORTRY Certificate Details) contains two object identifiers (OIDs) of interest within the key usage value:

Figure 4: Extended Key Usage

```
X509v3 Extended Key Usage:
                1.3.6.1.4.1.311.10.3.5, 1.3.6.1.4.1.311.10.3.5.1, Code Signing
```

RFC 5280 Section 4.2.1.12 defines Extended Key Usage (EKU). The EKU values in this signature help identify which method was used to sign this file and what purposes this signing certificate may be used for. The values defined show that this certificate is used in the Windows Hardware Compatibility driver signing process and is used specifically for attestation signed drivers. Table 1 shows the OID descriptions.

Table 2: WHQL Extended Key Usage object IDs

EKU OID	Symbolic Name	Description
1.3.6.1.4.1.311.10.3.5	szOID_WHQL_CRYPTO	Windows Hardware Driver Verification
1.3.6.1.4.1.311.10.3.5.1	szOID_ATTEST_WHQL_CRYPTO	Windows Hardware Driver Attested Verification

The connection between the POORTRY sample, the attestation certificate, and the numerous legitimate samples signed with this certificate led Mandiant to assess with high confidence that this malware was verified via the Windows Hardware Compatibility process.

RFC 2315 for the PKCS #7 v1.5 specification defines a SignerInfo content type, which for Authenticode signed PEs contains several interesting structures that can be used to identify samples related to the initially identified POORTRY driver (6fcf56f6ca3210ec397e55f727353c4a).

The field of interest, `programName`, is contained in the SpcSpOpusInfo attribute, which is specific to Authenticode. Mandiant assesses with high confidence that the `programName` field (hereafter referred to as Program Name) for attestation signed drivers contains identifiable information about the individual hardware vendor who submitted the driver for attestation signing.

Figure 5: Windows Authenticode Portable Executable Signature Format

```
SpcSpOpusInfo
  SpcSpOpusInfo is identified by SPC_SP_OPUS_INFO_OBJID
  (1.3.6.1.4.1.311.2.1.12) and is defined as follows:
  SpcSpOpusInfo ::= SEQUENCE {
    programName  [0] EXPLICIT SpcString OPTIONAL,
    moreInfo     [1] EXPLICIT SpcLink OPTIONAL,
  } --#public--

  SpcSpOpusInfo has two fields:
    programName
      This field contains the program description:
      If publisher chooses not to specify a description, the SpcString
      structure contains a zero-length program name.
      If the publisher chooses to specify a
      description, the SpcString structure contains a Unicode string.
    moreInfo
      This field is set to an SPCLink structure that contains a URL for
      a Web site with more information about the signer. The URL is an
      ASCII string.
```

Figure 6: Program Name value from POORTRY Authenticode data (6fcf56f6ca3210ec397e55f727353c4a)

大连纵梦网络科技有限公司

This field becomes an important artifact for identifying additional associated samples, and by pivoting on the Program Name, Mandiant identified eleven new suspicious files, including an additional POORTRY sample.

Table 3: Samples with Program Name of 大连纵梦网络科技有限公司

MD5	Family	Filename	Signature Date
6fcf56f6ca3210ec397e55f727353c4a	POORTRY	4.sys	2022/09/15 11:49
ee6b1a79cb6641aa44c762ee90786fe0	POORTRY	NodeDriver.sys	2022/08/21 01:43
1f2888e57fdd6aee466962c25ba7d62d		Air_SYSTEM10.sys	2022/10/01 11:43
22949977ce5cd96ba674b403a9c81285		PcieCubed.sys	2022/08/20 09:37
4e1f656001af3677856f664e96282a6f		Sense5Ext.sys	2022/08/09 07:20
7f9309f5e4defec132b622fadbcad511			2022/08/24 07:33
acac842a46f3501fe407b1db1b247a0b			2022/08/23 04:40
b164daf106566f444dfb280d743bc2f7			2022/08/17 10:48
bd25be845c151370ff177509d95d5add		2.sys	2022/09/19 24:33
dc564bac7258e16627b9de0ce39fae25		7.sys	2022/08/19 08:03

f9844524fb0009e5b784c21c7bad4220	Sense5Ext.sys	2022/08/22 14:48

The programName field for attestation signed drivers appears to be populated by the X.509 Subject Organization Name (O) of the EV Code Signing certificate used to sign the initial CAB submission to the WHCP portal. This is corroborated by the high amount of malicious detections for samples associated with this Organization Name and other corresponding Program Name values on VirusTotal and within other Mandiant data sets. At time of writing, we have not been able to confirm with Microsoft that this is the exact mechanism for how the programName field is populated for attestation signed drivers.

Table 4: Digicert EV code signing certificates with Organization Name of 大连纵梦网络科技有限公司

MD5	Family	Certificate Serial
05a56a88f34718cabd078dfd6b180ed0	Fast Reverse Proxy	01:15:3e:7a:3c:8d:c5:0b:3d:23:c8:ba:31:d3:70:52
2406150783d3ec5de13c2654db1a13d5	Fast Reverse Proxy	01:15:3e:7a:3c:8d:c5:0b:3d:23:c8:ba:31:d3:70:52
29506adae5c1e97de49e3a0d3cd974d4	Fast Reverse Proxy	01:15:3e:7a:3c:8d:c5:0b:3d:23:c8:ba:31:d3:70:52
48c1288cd35504de6f4bd97ec02decb1	Fast Reverse Proxy	01:15:3e:7a:3c:8d:c5:0b:3d:23:c8:ba:31:d3:70:52
578e70a8a7c1972bbc35c3e14e53cbee	Fast Reverse Proxy	01:15:3e:7a:3c:8d:c5:0b:3d:23:c8:ba:31:d3:70:52
6216fba5cf44aa99a73ca919301142e9	Fast Reverse Proxy	01:15:3e:7a:3c:8d:c5:0b:3d:23:c8:ba:31:d3:70:52
69fa8946c326d4b66a371608d8ffbe5e	Fast Reverse Proxy	01:15:3e:7a:3c:8d:c5:0b:3d:23:c8:ba:31:d3:70:52
6e4e37641e24edc89cfa3e999962ea34	Fast Reverse Proxy	0c:25:f1:f2:a8:d4:a2:93:21:e8:28:6e:ed:50:e3:e2
8a930742d1da0fcfe5492d4eb817727c	Fast Reverse Proxy	01:15:3e:7a:3c:8d:c5:0b:3d:23:c8:ba:31:d3:70:52
8fbad6e5aa15857f761e6a7a75967e85	SOGU Launcher	**03:25:0b:78:25:67:56:fc:10:db:c6:7a:22:52:7b:44**
976bac6cfb21288b4542d5afe7ce7be7	Fast Reverse Proxy	01:15:3e:7a:3c:8d:c5:0b:3d:23:c8:ba:31:d3:70:52
aaeedaa5880e38dc63a5724cf18baf13	Fast Reverse Proxy	01:15:3e:7a:3c:8d:c5:0b:3d:23:c8:ba:31:d3:70:52
ab5d85079e299ac49fcc9f12516243de	SOGU Launcher	**0c:59:d4:65:80:f0:39:af:2c:4a:b6:ba:0f:fe:d1:97**
c43de22826a424b2d24cf1b4b694ce07	SOGU Launcher	**0c:59:d4:65:80:f0:39:af:2c:4a:b6:ba:0f:fe:d1:97**
d312a6aeffec3cff78e9fad141d3aaba	Fast Reverse Proxy	01:15:3e:7a:3c:8d:c5:0b:3d:23:c8:ba:31:d3:70:52
d36084aad079ca8d91c2985eca80327b	Fast Reverse Proxy	01:15:3e:7a:3c:8d:c5:0b:3d:23:c8:ba:31:d3:70:52
e086d7d5a5657800a0d7e9c144fac16d	Fast Reverse Proxy	01:15:3e:7a:3c:8d:c5:0b:3d:23:c8:ba:31:d3:70:52

All the observed corresponding EV code signing certificates were issued by Digicert. Over time certificate serial 01:15:3e:7a:3c:8d:c5:0b:3d:23:c8:ba:31:d3:70:52 was revoked, however several others appear to have not been revoked (bolded in Table 4). These corresponding Extended Validation certificates were used to sign launchers for SOGU malware utilized by Temp.Hex as well as signed distributions of the open source Fast Reverse Proxy tool, which has been used by suspected Iranian state-sponsored threat actors in intrusions observed by Mandiant.

Utilizing the OIDs and certificate data, YARA rules were developed to collect additional attestation signed drivers.

Examining these additional attestation signed drivers led to 57 suspicious samples that shared program names that were observed in malicious binaries (Appendix B: Indicators of Interest). These samples were spread across nine different program names.

Figure 7: Identified Program Names on suspicious attestation signed samples

`Qi Lijun`

`Luck Bigger Technology Co., Ltd`

`XinSing Network Service Co., Ltd`

`Hangzhou Shunwang Technology Co.,Ltd`

福州超人

北京弘道长兴国际贸易有限公司

福建奥创互娱科技有限公司

厦门恒信卓越网络科技有限公司

大连纵梦网络科技有限公司

Malicious Driver Signing as a Service

The suspicious samples identified through this investigation have led to multiple development environment artifacts, specifically program database (PDB) paths, implying multiple different development environments and potentially multiple different malware authors.

Mandiant has previously observed scenarios when it is suspected that groups leverage a common criminal service for code signing. This is not a new phenomenon, and has been documented by the Certified Malware project at the University of Maryland in 2017. This is what Mandiant believes is occurring with these suspicious attestation signed drivers and related EV signed samples.

The use of stolen or fraudulently obtained code signing certificates by threat actors has been a common tactic and providing these certificates or signing services has proven a lucrative niche in the underground economy. Mandiant has identified numerous threat actors and services advertising in a variety of languages, including English, Russian, and Chinese, that claim to provide code signing certificates or sign malware on behalf of threat actors. For example, while analyzing chat messages leaked by the Twitter user "@ContiLeaks," Mandiant identified several instances where threat actors involved in Trickbot operations purchased code signing certificates from multiple threat actors, with observed pricing ranging between approximately $1,000-$3,000 USD for a single certificate.

While most of these advertisements only mention EV code signing certificates, we have identified a small number of discussions focused on signing drivers through WHQL. While most of these discussions lamented to the challenges presented by WHQL restrictions, we observed at least one actor who mentioned experience signing drivers with WHQL, and we have also identified multiple websites on the open Internet advertising WHQL driver signing services to enterprise businesses. While we are unable to link the signed payloads observed in this activity to any of the identified services, it's plausible that actors are either enlisting services from underground forums or abusing commercial services to obtain signed driver malware.

A pattern emerges of suspected malicious attestation signed drivers that contain the `programName` corresponding to EV certificates that have also signed other suspected malicious samples. The Certificates appear to be issued primarily via Digicert and Globalsign to Chinese customers, indicating possible abuse of a Chinese market certificate reseller or signing service.

Given the different company names identified and the differing development environments Mandiant suspects there is a service provider getting these malware samples signed through the attestation process on behalf of the actors. Unfortunately, at this time, this assessment is stated with low confidence.

Hunting and Blocking

Attestation signing is a legitimate Microsoft program, and the resulting drivers are signed with legitimate Microsoft certificates. This makes execution-time detection difficult as Microsoft and most EDR tools will allow Microsoft signed binaries to load. Organizations must instead depend on behavioral detections to overcome the implicit trust granted to Microsoft-signed binaries and alert on suspicious or rootkit-like activities. For proactive hunts, however, there are numerous ways to search for these files.

YARA Rules and Descriptions

M_Hunting_Signed_Driver_Attestation_1

The OLEs allow detection to be implemented to identify any binary that is signed via the attestation process. This rule matches on the presence of the OLEs and the Microsoft Windows Hardware Compatibility Publisher certificate subject.

M_Win_Hunting_CertEngine_Attestation_ProgramName_1

The identified company names that were in the certificate program name can be used to home in on potentially suspicious samples. However, know that due to the nature of these certificates it is not true that all samples with the certificate are malicious, but simply have been abused in the past and warrant further investigation.

M_CertEngine_Malicious_Attestation_Signed_Driver

The VirusTotal dataset has additional data available for access via LiveHunt rules. This includes various tags and other metadata from the related sandbox execution. This information can be used to identify suspected malicious attestation signed binaries by combining the M_Hunting_Signed_Driver_Attestation_1 rule with the malicious count metadata.

M_Hunting_Win_ConventionEngine_PDB_Attestation_Multiple_1

As documented in the Definitive Dossier of Devilish Debug Details, PDB paths can be used to identify strings that are present within the malware.However, it's important to remember that this is a consequence of the malware and malware developers, and not the certificate or signing process.

See Appendix A: YARA for the full list of detections.

Conclusion

The attestation signing process offloads the responsibility of verifying the identity of the requesting hardware or software vendor to the Certificate Authorities. In theory this is a valid process as the CAs must follow agreed upon procedures to verify the identity of the requesting entity and the authority of the individual making the request to represent the software vendor. However, this process is being abused to obtain malware signed by Microsoft.

This is not a new occurrence; both GData and BitDefender released reports on Microsoft signed malicious drivers in 2021. "Microsoft signed a malicious Netfilter rootkit" and "Digitally-Signed Rootkits are Back – A Look at FiveSys and Companions" discussed malicious drivers signed via the same attestation process discussed in this blog post.

While this blog post has focused on POORTRY and the attestation signing process, Mandiant has observed other malware being signed via attestation. TEMPLESHOT is a malware family consisting of dropper, backdoor, a filter driver, and a protection driver. The TEMPLESHOT driver with MD5 `48bf11dd6c22e241b745d3bb1d562ca1` has been observed in the wild and is signed via attestation.

Acknowledgements

Use of the Signify python library made automated analysis of Authenticode data extremely efficient. This content would not have been possible without the assistance of analysts across the Mandiant Intelligence and FLARE organizations.

Appendix A: YARA

```
import "pe"

rule M_Hunting_Signed_Driver_Attestation_1

{

    meta:

        author = "Mandiant"

        date_created = "2022-10-20"

        description = "Find driver signed via Microsoft attestation signing only (no EV certificate signing outside of
Microsoft Windows Hardware Compatibility Publisher)" //https://learn.microsoft.com/en-us/windows-
hardware/drivers/dashboard/code-signing-attestation

    strings:

        $whql_oid = {2b0601040182370a030501} //OID 1.3.6.1.4.1.311.10.3.5.1, Windows Hardware Quality Labs (WHQL)
crypto -- "szOID_WHQL_CRYPTO"

        $spc_statement_type = {2b060104018237020115} //OID 1.3.6.1.4.1.311.2.1.21, SPC_INDIVIDUAL_SP_KEY_PURPOSE_OBJID

        $spc_sp_opus_info_oid = {2b06010401823702010c} //OID 1.3.6.1.4.1.311.2.1.12, SPC_SP_OPUS_INFO_OBJID

    condition:

        pe.signatures[0].subject == "/C=US/ST=Washington/L=Redmond/O=Microsoft Corporation/CN=Microsoft Windows
Hardware Compatibility Publisher" and

        $whql_oid and

        $spc_sp_opus_info_oid and

        $spc_statement_type

}
```

```
import "pe"

rule M_Win_Hunting_CertEngine_Attestation_ProgramName_1

{

    meta:

        author = "Mandiant"

        description = "Find driver signed via Microsoft attestation signing only with one of the identified company
names of interest."

    strings:

        $whql_oid = {2b0601040182370a030501} //OID 1.3.6.1.4.1.311.10.3.5.1, Windows Hardware Quality Labs (WHQL)
crypto -- "szOID_WHQL_CRYPTO"

        $spc_statement_type = {2b060104018237020115} //OID 1.3.6.1.4.1.311.2.1.21, SPC_INDIVIDUAL_SP_KEY_PURPOSE_OBJID

        $spc_sp_opus_info_oid = {2b06010401823702010c} //OID 1.3.6.1.4.1.311.2.1.12, SPC_SP_OPUS_INFO_OBJID

        $unicode1 = {59278FDE 7EB568A6 7F517EDC 79D16280 67099650 516C53F8}

        $unicode2 = {51 69 20 4c 69 6a 75 6e}

        $unicode3 = {4c 75 63 6b 20 42 69 67 67 65 72 20 54 65 63 68 6e 6f 6c 6f 67 79 20 43 6f 2e 2c 20 4c 74 64}

        $unicode4 = {58 69 6e 53 69 6e 67 20 4e 65 74 77 6f 72 6b 20 53 65 72 76 69 63 65 20 43 6f 2e 2c 20 4c 74 64}

        $unicode5 = {48 61 6e 67 7a 68 6f 75 20 53 68 75 6e 77 61 6e 67 20 54 65 63 68 6e 6f 6c 6f 67 79 20 43 6f 2e
2c 4c 74 64}

        $unicode6 = {54 41 20 54 72 69 75 6d 70 68 2d 41 64 6c 65 72 20 47 6d 62 48}

        $unicode7 = {798f 5dde 8d85 4eba}

        $unicode8 = {5317 4eac 5f18 9053 957f 5174 56fd 9645 8d38 6613 6709 9650 516c 53f8}

        $unicode9 = {798f 5efa 5965 521b 4e92 5a31 79d1 6280 6709 9650 516c 53f8}

        $unicode10 = {53a6 95e8 6052 4fe1 5353 8d8a 7f51 7edc 79d1 6280 6709 9650 516c 53f8}

    condition:

        $whql_oid and

        $spc_sp_opus_info_oid and

        $spc_statement_type and

        pe.signatures[0].subject == "/C=US/ST=Washington/L=Redmond/O=Microsoft Corporation/CN=Microsoft Windows
Hardware Compatibility Publisher" and

        (1 of ($unicode*))

}
```

```
import "vt"

import "pe"

rule M_CertEngine_Malicious_Attestation_Signed_Driver

{

    meta:

        author = "Mandiant"

        description = "Find driver signed via Microsoft attestation signing only and greater than 3 malicious hits in
VirusTotal.”

    strings:

        $whql_oid = {2b06010401823 70a030501} //OID 1.3.6.1.4.1.311.10.3.5.1, Windows Hardware Quality Labs (WHQL)
crypto -- "szOID_WHQL_CRYPTO"

        $spc_statement_type = {2b060104018237020115} //OID 1.3.6.1.4.1.311.2.1.21, SPC_INDIVIDUAL_SP_KEY_PURPOSE_OBJID

        $spc_sp_opus_info_oid = {2b06010401823702010c} //OID 1.3.6.1.4.1.311.2.1.12, SPC_SP_OPUS_INFO_OBJID

    condition:

        for any tag in vt.metadata.tags : ( tag == "signed" ) and

        pe.signatures[0].subject == "/C=US/ST=Washington/L=Redmond/O=Microsoft Corporation/CN=Microsoft Windows
Hardware Compatibility Publisher" and

        vt.metadata.analysis_stats.malicious > 3 and

        $whql_oid and

        $spc_sp_opus_info_oid and

        $spc_statement_type

}


rule M_Hunting_Win_ConventionEngine_PDB_Attestation_Multiple_1

{

    meta:

        author = "Mandiant"

        description = "Looking for PDB path strings that has been observed in malicious samples which were attestation
signed"

    strings:

        $anchor = "RSDS"

        $pdb1 = /RSDS[\x00-\xFF]{20}[a-zA-Z]:\\.{0,250}gamehacks.{0,250}boot_driver.{0,250}\.pdb\x00/ nocase

        $pdb2 = /RSDS[\x00-\xFF]{20}[a-zA-Z]:\\.{0,250}MyDriver1.{0,250}wfp_vpn.{0,250}\.pdb\x00/ nocase

        $pdb3 = /RSDS[\x00-\xFF]{20}[a-zA-Z]:\\.{0,250}FilDriverx64_win10.{0,250}\.pdb\x00/ nocase

        $pdb4 = /RSDS[\x00-\xFF]{20}[a-zA-Z]:\\.{0,250}RedDriver_win10.{0,250}\.pdb\x00/ nocase

        $pdb5 = /RSDS[\x00-\xFF]{20}[a-zA-Z]:\\.{0,250}sellcode.{0,250}MyDriver.{0,250}\.pdb\x00/ nocase

        $pdb6 = /RSDS[\x00-\xFF]{20}[a-zA-Z]:\\.{0,250}Users\\ljl11{0,250}\.pdb\x00/ nocase

        $pdb7 = /RSDS[\x00-\xFF]{20}[a-zA-Z]:\\.{0,250}RkDriver64.{0,250}MyDriver1.{0,250}\.pdb\x00/ nocase

        $pdb8 = /RSDS[\x00-\xFF]{20}[a-zA-Z]:\\.{0,250}\\ApcHelper.{0,250}TSComputerManager.{0,250}\.pdb\x00/ nocase

    condition:

        (uint16(0) == 0x5A4D) and uint32(uint32(0x3C)) == 0x00004550 and filesize < 20MB and $anchor and (1 of
($pdb*))

}
```

Appendix B: Indicators of Interest

Attestation Signed Binaries with Suspicious Program Name Values

This table is sorted by Signature Date. Signature Date is an authenticated attribute, containing the timestamp of signing. Sorting by this date allows readers to view how the `programName` is used and changed over time.

One sample (`688c138fffbb4e7297289433c79d62f5`) does not have a Signature Date, and this is likely due to binary tampering including the use of VMProtect after signing and other modifications.

MD5	Program Name	Signature Date
688c138fffbb4e7297289433c79d62f5	北京弘道长兴国际贸易有限公司	N/A
0b4a0fe7db8400ef65ce7618177351cf	福建奥创互娱科技有限公司	2021/07/09 11:35
6e3516775e7e009777dcdb7a314f1482	福建奥创互娱科技有限公司	2021/07/19 07:39
ea5f6ab5666193f805d13a49009f0699	福建奥创互娱科技有限公司	2021/07/20 06:43
63960dbc7d63767edb6e1e2dc6f0707b	福建奥创互娱科技有限公司	2021/07/28 13:05
ddee86b84dcb72835b57b1d049e9e0cd	福建奥创互娱科技有限公司	2021/07/29 09:25
19d99758b1f33b418cb008530b61a1e7	福建奥创互娱科技有限公司	2021/07/29 10:02
f9aad310a5d5c80bbc61d10cc797e4f0	北京弘道长兴国际贸易有限公司	2021/11/06 17:38
ff43f91f2465504e5e67d0b37d92ef18	厦门恒信卓越网络科技有限公司	2021/12/30 06:12
45be5c0e7dfe37f88f1fa6c2fbb462c5	厦门恒信卓越网络科技有限公司	2022/01/13 24:00
26d6833b1875b138ea34d6ab430cafcd	厦门恒信卓越网络科技有限公司	2022/02/07 03:47
561bc6902367d9e43e27c5543e7a5818	厦门恒信卓越网络科技有限公司	2022/02/09 11:35
929b293090bcc7900c1e8f9ba519e219	厦门恒信卓越网络科技有限公司	2022/02/13 12:25
b500ee8d8cb045936d2996a1747bcded	厦门恒信卓越网络科技有限公司	2022/02/14 24:25
42200c8422347f63b3edb45ea5aa9c45	厦门恒信卓越网络科技有限公司	2022/02/14 12:25
48fc05c42549d0b3ec9e73bbb5be40dc	厦门恒信卓越网络科技有限公司	2022/02/14 12:25
bf13a2f4e2deb62b7dee98a012e94d61	厦门恒信卓越网络科技有限公司	2022/02/14 12:25
d66fc4e2f537566bb4d91cdea0ac64e5	厦门恒信卓越网络科技有限公司	2022/02/14 12:25
de4b5043c82ab3b36b4ae73a2e96d969	厦门恒信卓越网络科技有限公司	2022/02/14 12:25
cc29cf2294175315acbf33054151f3cd	厦门恒信卓越网络科技有限公司	2022/02/15 06:07
6e730cf4ebcd166d26414378cab3a6d8	厦门恒信卓越网络科技有限公司	2022/02/18 06:58
8e4d0f679b092296a2f74cf812907d05	厦门恒信卓越网络科技有限公司	2022/02/18 06:58
f8ccabcbe08bbd2c8420f4d1cffcefd8	厦门恒信卓越网络科技有限公司	2022/02/18 06:58

9f1d3b0fb49e063f4804aa60b7b708ac	厦门恒信卓越网络科技有限公司	2022/02/18 08:23
2bbfb9cb4550109da5ae336d3d3dd984	厦门恒信卓越网络科技有限公司	2022/02/23 03:55
42a417e54639c69f033f72bbafe6e09a	北京弘道长兴国际贸易有限公司	2022/02/25 09:18
7ee0c884e7d282958c5b3a9e47f23e13	北京弘道长兴国际贸易有限公司	2022/02/26 24:58
66c1452335767660136880880b03103e3	厦门恒信卓越网络科技有限公司	2022/03/08 07:16
1f929fd617471c4977b522c71b4c91ed	北京弘道长兴国际贸易有限公司	2022/03/26 24:09
4a0f22286134a58d9d20f911a608f636	福州超人	2022/03/28 09:34
947ebc3f481a7b9ee3cf3a34d9830159	福州超人	2022/03/28 09:40
33b5485b35b33fd8ead5a38899522cce	福州超人	2022/03/28 10:20
721b40a0c2a0257443f7dcc2c697e28a	福州超人	2022/04/09 17:06
b44dfd8c5e7b0c8652d7a647dfe252e4	福州超人	2022/05/03 09:25
1a57c1d80018bfef1e243f9eba2955f2	北京弘道长兴国际贸易有限公司	2022/05/09 01:18
ac2a1f2ae6b547619bef93dfadb48937	福州超人	2022/05/19 07:09
8ac6ef2475ec89d3709fc124573cb380	北京弘道长兴国际贸易有限公司	2022/05/31 11:06
b34403502499741762912c7bfc9ff21f	Hangzhou Shunwang Technology Co.,Ltd	2022/06/13 08:25
734b3a6e6cbd1f53fbb693140d2c3049	北京弘道长兴国际贸易有限公司	2022/06/13 08:45
c0471f78648643950217620f6e7e24cc	北京弘道长兴国际贸易有限公司	2022/06/13 08:45
228f9f0a0466fba21ac085626020a8e1	Qi Lijun	2022/08/02 16:10
65a3f812ea031f4d53ba09f33c058ab6	Qi Lijun	2022/08/02 16:10
7d78b5773845c5189ca09227d27a9d5a	Qi Lijun	2022/08/03 01:56
e7ff38a94ad765eb305fc7f0837f5913	Qi Lijun	2022/08/03 01:58
4e1f656001af3677856f664e96282a6f	大连纵梦网络科技有限公司	2022/08/09 07:20
b164daf106566f444dfb280d743bc2f7	大连纵梦网络科技有限公司	2022/08/17 10:48
dc564bac7258e16627b9de0ce39fae25	大连纵梦网络科技有限公司	2022/08/19 08:03
22949977ce5cd96ba674b403a9c81285	大连纵梦网络科技有限公司	2022/08/20 09:37
ee6b1a79cb6641aa44c762ee90786fe0	大连纵梦网络科技有限公司	2022/08/21 01:43

f9844524fb0009e5b784c21c7bad4220	大连纵梦网络科技有限公司	2022/08/22 14:48
acac842a46f3501fe407b1db1b247a0b	大连纵梦网络科技有限公司	2022/08/23 04:40
7f9309f5e4defec132b622fadbcad511	大连纵梦网络科技有限公司	2022/08/24 07:33
7ba744b584e28190eb03b9ecd1bb9374	XinSing Network Service Co., Ltd	2022/09/07 02:24
6fcf56f6ca3210ec397e55f727353c4a	大连纵梦网络科技有限公司	2022/09/15 11:49
bd25be845c151370ff177509d95d5add	大连纵梦网络科技有限公司	2022/09/19 24:33
1f2888e57fdd6aee466962c25ba7d62d	大连纵梦网络科技有限公司	2022/10/01 11:43
909f3fc221acbe999483c87d9ead024a	Luck Bigger Technology Co., Ltd	2022/10/19 13:15

Signed POORTRY Samples

The following table includes signed POORTRY samples.

Compile Time	Signing Status	Signing Time	PDB path	MD5	Filename	Serial	C N
20220602 10:09:08	Revoked	20220811 13:27:00	D:\KApcHelper\x64\ Release\KApcHelper.pdb	10f3679384a03cb4 87bda9621ceb5f90	prokiller64.sys	62:7d:fd:f7:3a:14:55:de: 51:43:a2:70:79:9e:6b:7b	Zl l: T C
20220602 10:09:08	Revoked		D:\KApcHelper\x64\ Release\KApcHelper.pdb	04a88f5974caa621 cee18f34300fc08a	gftkyj64.sys	62:7d:fd:f7:3a:14:55:de: 51:43:a2:70:79:9e:6b:7b	Zl l: T C
20220602 10:09:08		20220915 15:49:00		6fcf56f6ca3210ec 397e55f727353c4a		33:00:00:00:57:ee:4d:65:9a:9 2:3e:7c:10:00:00:00:00:00:57	M W Ha C Pu
20220606 15:14:46	Expired		D:\KApcHelper\x64\ Release\KApcHelper.pdb	0f16a43f79890346 41fd2de3eb268bf1	KApcHelper_x64.sys	43:bb:43:7d:60:98:66:28: 6d:d8:39:e1:d0:03:09:f5	NV C
20220820 15:19:01		20220821 05:43:00		ee6b1a79cb6641aa 44c762ee90786fe0	NodeDriver.sys	33:00:00:00:57:ee:4d:65:9a:9 2:3e:7c:10:00:00:00:00:00:57	M W Ha C Pu
20221002 19:48:02		20221019 17:15:00		909f3fc221acbe99 9483c87d9ead024a	LcTkA.sys	33:00:00:00:57:ee:4d:65:9a:9 2:3e:7c:10:00:00:00:00:00:57	M W Ha C Pu

Extended Validation Signed Samples

The following table includes samples signed by EV certificates where the Organization Name is 大连纵梦网络科技有限公司.

Compile Time	Signed Time	MD5	Family	Filename	Certificate Serial	Certificate Issuer Common Name	Organization Name

19700101 00:00:00	20201006 16:26:00	05a56a88f34718ca bd078dfd6b180ed0	Fast Reverse Proxy	frpc.exe	01:15:3e:7a:3c:8d:c5:0b: 3d:23:c8:ba:31:d3:70:52	DigiCert EV Code Signing CA	大连纵梦网络科技有限公司
19700101 00:00:00	20201128 18:12:00	2406150783d3ec5d e13c2654db1a13d5	Fast Reverse Proxy	frpc.exe	01:15:3e:7a:3c:8d:c5:0b: 3d:23:c8:ba:31:d3:70:52	DigiCert EV Code Signing CA	大连纵梦网络科技有限公司
19700101 00:00:00	20210226 22:11:00	29506adae5c1e97d e49e3a0d3cd974d4	Fast Reverse Proxy	%home%\unpack\ sakuralauncher_v2.0.1.2 \frpc.exe	01:15:3e:7a:3c:8d:c5:0b: 3d:23:c8:ba:31:d3:70:52	DigiCert EV Code Signing CA	大连纵梦网络科技有限公司
19700101 00:00:00	20220219 13:29:00	48c1288cd35504de 6f4bd97ec02decb1	Fast Reverse Proxy	svchost.exe	01:15:3e:7a:3c:8d:c5:0b: 3d:23:c8:ba:31:d3:70:52	DigiCert EV Code Signing CA	大连纵梦网络科技有限公司
19700101 00:00:00	20200820 12:34:00	578e70a8a7c1972b bc35c3e14e53cbee	Fast Reverse Proxy	frpc.exe	01:15:3e:7a:3c:8d:c5:0b: 3d:23:c8:ba:31:d3:70:52	DigiCert EV Code Signing CA	大连纵梦网络科技有限公司
19700101 00:00:00	20201128 18:13:00	6216fba5cf44aa99 a73ca919301142e9	Fast Reverse Proxy		01:15:3e:7a:3c:8d:c5:0b: 3d:23:c8:ba:31:d3:70:52	DigiCert EV Code Signing CA	大连纵梦网络科技有限公司
19700101 00:00:00	20220219 13:29:00	69fa8946c326d4b6 6a371608d8ffbe5e	Fast Reverse Proxy	frpc_windows_amd64.exe	01:15:3e:7a:3c:8d:c5:0b: 3d:23:c8:ba:31:d3:70:52	DigiCert EV Code Signing CA	大连纵梦网络科技有限公司
19700101 00:00:00	20200802 07:11:00	6e4e37641e24edc8 9cfa3e999962ea34	Fast Reverse Proxy	frpc.exe	0c:25:f1:f2:a8:d4:a2:93: 21:e8:28:6e:ed:50:e3:e2	DigiCert EV Code Signing CA	大连纵梦网络科技有限公司
19700101 00:00:00	20210605 19:09:00	8a930742d1da0fcf e5492d4eb817727c	Fast Reverse Proxy	c:\program files\sakurafrplauncher \frpc.exe	01:15:3e:7a:3c:8d:c5:0b: 3d:23:c8:ba:31:d3:70:52	DigiCert EV Code Signing CA	大连纵梦网络科技有限公司
20211220 07:37:56		8fbad6e5aa15857f 761e6a7a75967e85	SOGU Launcher	powerdvd18.exe	03:25:0b:78:25:67:56:fc: 10:db:c6:7a:22:52:7b:44	DigiCert EV Code Signing CA	大连纵梦网络科技有限公司
19700101 00:00:00	20201224 19:02:00	976bac6cfb21288b 4542d5afe7ce7be7	Fast Reverse Proxy	frpc.exe	01:15:3e:7a:3c:8d:c5:0b: 3d:23:c8:ba:31:d3:70:52	DigiCert EV Code Signing CA	大连纵梦网络科技有限公司
19700101 00:00:00	20210605 19:09:00	aaeedaa5880e38dc 63a5724cf18baf13	Fast Reverse Proxy	frpc_windows_386.exe	01:15:3e:7a:3c:8d:c5:0b: 3d:23:c8:ba:31:d3:70:52	DigiCert EV Code Signing CA	大连纵梦网络科技有限公司
20200704 03:53:04	20200704 08:13:00	ab5d85079e299ac4 9fcc9f12516243de	SOGU Launcher	SmadavMain.exe	0c:59:d4:65:80:f0:39:af: 2c:4a:b6:ba:0f:fe:d1:97	DigiCert High Assurance Code Signing CA-1	大连纵梦网络科技有限公司

20200522 10:23:03	20200523 06:16:00	c43de22826a424b2 d24cf1b4b694ce07	SOGU Launcher	AdobeHelp.exe	0c:59:d4:65:80:f0:39:af: 2c:4a:b6:ba:0f:fe:d1:97	DigiCert High Assurance Code Signing CA-1	大连纵梦网络科技有限公司
19700101 00:00:00	20201006 16:28:00	d312a6aeffec3cff 78e9fad141d3aaba	Fast Reverse Proxy	frpc.exe	01:15:3e:7a:3c:8d:c5:0b: 3d:23:c8:ba:31:d3:70:52	DigiCert EV Code Signing CA	大连纵梦网络科技有限公司
19700101 00:00:00	20210321 09:12:00	d36084aad079ca8d 91c2985eca80327b	Fast Reverse Proxy	c:\program files\sakurafrplauncher \frpc.exe	01:15:3e:7a:3c:8d:c5:0b: 3d:23:c8:ba:31:d3:70:52	DigiCert EV Code Signing CA	大连纵梦网络科技有限公司
19700101 00:00:00	20201224 19:02:00	e086d7d5a5657800 a0d7e9c144fac16d	Fast Reverse Proxy	frpc.exe	01:15:3e:7a:3c:8d:c5:0b: 3d:23:c8:ba:31:d3:70:52	DigiCert EV Code Signing CA	大连纵梦网络科技有限公司

Suspicious Attestation Signed Samples

The following list of MD5s are attestation signed binaries that have been identified as suspicious by numerous security solutions. While each one may not be directly malicious, they warrant an investigation should they be present in an environment.

0080fde587d6aedccb08db1317360d32	ff985a86bfa60576a8e86b05603ac5fa	b00c95692923b8c1e2d45c4a64a5ff05
00a7538086c266e8bcf8a0b1c2b6a2e4	62f289f3b55b0886c419a5077d11eb3c	b0fea98c70e510f88b57f45a3f516326
00dd476fa04da76fc2ed37cfdde59875	63960dbc7d63767edb6e1e2dc6f0707b	b164daf106566f444dfb280d743bc2f7
024e92733def0b1180f0ee54b81e5836	63d877650a3219f5991fd66bafc46bc5	b34403502499741762912c7bfc9ff21f
03710450e5bebd207bbe471c4685dc49	64a81238d20dcbd4b21abb609040f698	b44dfd8c5e7b0c8652d7a647dfe252e4
07bac50f875f09ad644827c8918e6837	66c1452335767660136880 88b03103e3	b500ee8d8cb045936d2996a1747bcded
07c4309678ce891fdd868e10c6e7aad4	66d2860a078fb11832ceef28b23481c2	b5c73db8e70d6f46ad9b693f3ce060d2
0ae78b90151ec2b0457bb0c2675048f5	67ff9de8e72c4dfdf4b4404abf253e7e	b7239e06bcbe6e2c7bb2f7a859cbf4f7
0b4a0fe7db8400ef65ce7618177351cf	688c138fffbb4e7297289433c79d62f5	b83d8761748abb032ab5ae75519eaf71
0d0ffa28823276732a9e4dea5c25cc34	688ca3c12b63fec9f921334d24cf6f78	b849deae20052d72c3c623660fa97e64
14a1d3e07520df607635a3356877f5b9	6916b29893f618ba76b36bd8c297b7ac	b8783155d6be5bb3a6d75edaa7ae7f71
14e6507566a404e3158b3e36314bb3a1	6a066d2be83cf83f343d0550b0b8f206	b9d40581ae936662c37f2edc979d7e99
1548b70d8581cbde703b1fb50b48a6a8	6a23d752fbc30e603bbb050a83a580eb	ba9907be3a0752369082199ed126f8d8
163118c947aacd0978ad3e019c7d121f	6a893aab7b79b73da7a049c2707aabf1	bb46eb379caae3b05e32d3089c0dd6d0
179ca82f2e523be47df0dcebe808408d	6b0a733568d80be653fc9a568cdd88c5	bd25be845c151370ff177509d95d5add
198877a8ce99289f7281b1475c13ba9f	6c3180163e4a5371647e734c7c817de5	bf13a2f4e2deb62b7dee98a012e94d61
19d14bf80b3dc4e5b774b362f079a102	6c7479b5bb27f250fa32331b6457883a	c0471f78648643950217620f6e7e24cc

19d99758b1f33b418cb008530b61a1e7	6d32d2d7a44584c92115ac2a2c3ba3af	c0debd2cfb62fc2c56bfd4104b1ff760
1e63ec5b89edb805956f347b5b5cfaae	6e1bb443369973923c8eced16fcbd5cf	c12d465743b9c167fc819b7872cd014c
1f2888e57fdd6aee466962c25ba7d62d	6e3516775e7e009777dcdb7a314f1482	c35e6a0e1aef31ed9855499df4317acd
1f46065ac9479253e4babc42b72bc4a8	6e730cf4ebcd166d26414378cab3a6d8	c5120095bf08655407c2f0215d10ac1d
1f929fd617471c4977b522c71b4c91ed	6fcf56f6ca3210ec397e55f727353c4a	c77e931a6388b2040cc7c5a1a0f56d93
207cfc647647419adcfcc44c6059a1d1	7182ed3da406ba19bb9ffd8e4948d858	c7850060cfe574a2ef278ba46a136a5e
20f94c9cfc3cf012bf90546985f9f3c4	721b40a0c2a0257443f7dcc2c697e28a	c812fa7c628c3e19a3da5910acf6206e
22519936cd9e8c7d524b0590826c3e6e	72dbbd1dd61c6b0c2571e83f2c3d1825	c8495649615bf1b9f839d7f357d6d02f
228f9f0a0466fba21ac085626020a8e1	734b3a6e6cbd1f53fbb693140d2c3049	cadc3e4090aed708526f0d6016aba7fd
22949977ce5cd96ba674b403a9c81285	761939b0e442821985ab3281f97e6ceb	cb68b7979bbb55bbde0a8c60fe3e5184
232b0156173a9f8f5db6b65aa91e923b	76c6ae0157ea7f41f55ed7e7d241f910	cb6a416204b57470fab0b944d7b59756
23cebc6b0eb76262d796577895f418d2	7737e5e40a439899f326279b7face22c	cbc3d1c88a5d0491b7b50bb77ada93fe
24eb9eef69475e4980a555898b25f0c1	77392be5eae901ae371c37861aa88589	cc29cf2294175315acbf33054151f3cd
262c92f2437c80adf232ef147ca2d734	787782e0395b3d5e32cda6fdea2faba0	cd4b6d8bb762c2281c9b1142588ede4c
267c30e484322ad31fa9e1374d6653f0	79ebae9ab3f3b59c754ab1cc82bf7e95	ce455358bf71c88b45fcb5789100969a
26caf3361ec353593f51ebbd3fe5bbde	7a5896673b81beb5589b512c6d781a85	ce4d3a69331ff87920c903a4e4091904
26d6833b1875b138ea34d6ab430cafcd	7a9df5c46c7c65b807f78c6c0bb2c38c	ce658935ef6e223893121dce22908655
2739311a6bb1a7b0b88ff24bf603a54d	7b6e3fe75c5ae68d7d5a3ae7b00097e0	ce6ef4dc1dd54baddaa51eaf594a496a
27bb03f2659cd95bf9e7af899ee32728	7ba744b584e28190eb03b9ecd1bb9374	d11b9a4664ea03dfe3e8e1d737cd15f8
286b10451fe364310f4a7baeb0e94a3f	7c6c1b7e6378b4c0bcceee84e0e26fde	d22a56e31b4e1fd5b06d46fa56f59151
2a12b959c55f4a2d34f96e45e2417a71	7cb012393114dfb35d60e70166a97986	d27fac80339ad1f2ee86374884996c52
2aa8dc7a5dff7817ce0a9c7cf30847bf	7d78b5773845c5189ca09227d27a9d5a	d2ed678542a5d1db494dc47359861467
2bec13be352db14fc9665ddf128deb8c	7dd800f100a049a72983dd75f5286d70	d47494b717c82eca8278dea610e1265d
2cc14f20cf6847a2084f2c9cc0622015	7e0a6a234a64350e684544e272c7fc41	d60d8f3f12550dca4ba07ff61263b67f
2d84c734d813af49cec3c3aa4aa4e6e3	7e2e29707e7a601e8ea7f3e2f4d672a2	d60e235b769cadbc7e83090b79b73ed3
2e323c67a8781531a294684f7d2761ec	7e7002dc10c62fb674a3184f4ad6688a	d617c9a86328921a8caf924575faf2a2
2f6daca66d2f64c7b1b6f8693ea09cb7	7ee0b286003dc9e8006c22dcd70663f0	d66fc4e2f537566bb4d91cdea0ac64e5
309f16f50e9074ce797eb38eda279298	7f9309f5e4defec132b622fadbcad511	d6b2947d8ff985fa84d697cc6cfdb7ff

331113d1d54a3610f9c9bd72fc783721	811f8d76ff00c9eda27b51a0fb2b0d39	d6e506a1e0417c4507a5314529d84e34
33b5485b35b33fd8ead5a38899522cce	822bbdec4e5630c3170ee05119dcfb5c	d77209a21352486435d85e339596eeae
3452586b669e12c1c4ee9db3c1006018	8264b3bdf46c0ece4f66151a613baed5	d87f08d1e50f2a3423813bf161b40859
35c95b6b5f4a6a0bda56276846dae17b	832fe73a91993b387f9a49fafb9d4ea7	dc170d9bba14b0421c2514465055a93f
35deaa9d004714dc6ef9661b91889148	84ce2a917e3d4aefcfc7d17e4a840a99	dc564bac7258e16627b9de0ce39fae25
3608b3a24736dea4bf24a8ac5ae00e30	85063d67203b91bef9772446a1723021	dd1a5bd34f8cfa56e439c6fb275356d6
37d4ba16136986bfded2b6fc698abf02	860f5812d65dc157a59c14e57bc0eaaf	ddee86b84dcb72835b57b1d049e9e0cd
395ea8b7d0f257850a3a04a1484bac4d	8986b5b6013cfb2bd3e6c8d22c453390	de4b5043c82ab3b36b4ae73a2e96d969
398384a6cf2b7e26947d2e0acbfeeda5	8ac6ef2475ec89d3709fc124573cb380	e051141b1dcb9e7f889fea7c8b1d6ba5
39ee31f03fe1bb93d47f560f73deffa9	8af6a129902a594ddaceafba38b7c060	e0e0c46ba4f969919e2879717c60ef2a
3d4b685dcaebc5bba5f9421572a4ab91	8b423e0395ba6419fcedc0701327c97c	e2465ea5c2d5dac4ae1b8d50da1d7cce
3db8146544ee26866a8e99bacb11188c	8d38a092ae5a3511bedadb7243a84409	e2c146a2522e4f40e5036c3fe12c3560
3ecaf3ba4e93916714cc43320f6f2c58	8e4d0f679b092296a2f74cf812907d05	e30830c05ed3d2a3178a3678f3169bec
3fd815ebb7d2ab2b62cff3c777b51e30	8fc8c6e1b2a1047752f60549878fb55f	e5f62ef06b0dd656e1e47913f01f9f8a
4070a8b16f318d108be0984e628421ad	909f3fc221acbe999483c87d9ead024a	e6960ae657786979493da1786191bcf4
40fda9a3c1be41be414f3795b25647f5	90affc996a2932cb0fec4e31cd673ae9	e777e5a8d2ba97c82128f04272e7841c
415240633837ebcbd80e080ba99c03a9	90b9a4328c4f712815760f9da49bcb6a	e7ff38a94ad765eb305fc7f0837f5913
42200c8422347f63b3edb45ea5aa9c45	913d50851abf337abc3c73f2d4e7fb34	ea033ee6df904d863448ffef6386b6ae
42a417e54639c69f033f72bbafe6e09a	929b293090bcc7900c1e8f9ba519e219	ea45419d992c15002c93067840568121
4349378822e2316f18784c10c7ca08a1	934d0cda4cba428e9b75ff16d5f4b0b1	ea5f6ab5666193f805d13a49009f0699
45991757d4ca2dab9e81f2fcbbc1ae23	93c5faf90bc889963f10c608cbde5a14	ee3bad1f5508e2129e0b423b009383e3
45be5c0e7dfe37f88f1fa6c2fbb462c5	947ebc3f481a7b9ee3cf3a34d9830159	ee6b1a79cb6641aa44c762ee90786fe0
467e60b9a0d1153057e0cfd0e721e198	95a04866e6afb8e9b0426f5890681f9a	f07506c30237c96e49eecafa0e5a4ed4
48190fd615dcea5c6679b8e30a8bfec0	9885d56d64ac2391a43f02abb2202181	f111bd9b8e55f60f909649820e116430
486b1afce3484a784a1662513ca1272a	9a8323bc7187441a0d85b9a2e8f580e3	f35a8a8f36c13769b9e9fff05fa4f720
48bf11dd6c22e241b745d3bb1d562ca1	9c4034691f6508e2361b6fca890671f9	f4ee6bee04b2ed18024e3a64a0d58385
48fc05c42549d0b3ec9e73bbb5be40dc	9d1424c87d89095e3cd6785adb54d2ec	f59a1409ce773658e72ad73424841890
4a0f22286134a58d9d20f911a608f636	9dabf30a780794200cd068b145730317	f783277840bbd2023879a87d0788f36e

4b2e59a821589ab091a63770f4a658ed	9e91e55c89f9c17c0a2acaf4376cd72b	f78915cbf89d8749a0a4ab18a2b182bd
4d4c17d8b52cd89da0b17cc9653b2010	9f1d60d3cddea7f7558fad0217759094	f8ccabcbe08bbd2c8420f4d1cffcefd8
4d947e4163e8aeafbfc626eb033bc665	a0fdc4543687a1b341b365d6dd16551c	f9844524fb0009e5b784c21c7bad4220
4e1f656001af3677856f664e96282a6f	a2ee1cc9e80390ca248863004adbde60	f9aad310a5d5c80bbc61d10cc797e4f0
4e8d5c44bfdeffd0168f8a05f6a04e8b	a2f3bce86beef23aede69396dcf7e184	fa00cc96c5bea2979a59d0da0d22c83d
4f5c7367f2ebae0097b6f2f1bebd19b6	a55cb8be2887e99b4f662fc1ae08d265	fa914061f5a40b324454d3fb9fc85ca5
508d42f26f8bd562728e6fca866e05eb	a7251aad1e81c6194b34dabf6edd6b4a	faa5806826ff1ba749b70de0e14835c3
50d13758b811c794bc13769ee3b42e85	a9541530619a3ac2615b92603b705fe6	fbd9ba2b8b2d677d41c30a01c02cfd01
52494f624378ef6ee298f0fc73082d0e	aba1be25da0691761f593725e9c067e5	fd3b7234419fafc9bdd533f48896ed73
52fc9ec7a5c177fe27fb00b6c2c5ff09	ac2a1f2ae6b547619bef93dfadb48937	fd4cee1c7b8167f25a8b4b864ede3c5d
548d48b658305ffb77cc814ea080b542	ac7f0fcb6040eb47ea9855d418c32510	fdb6dae1e8c182089fdb86996436330c
561bc6902367d9e43e27c5543e7a5818	acac842a46f3501fe407b1db1b247a0b	fe2f8e46ae540d7299c61ba083d52399
5800a88d39fdf63e5a43bfcc6700d907	adab615712eac2719691d01b69254f29	fe7ecd399eec7036a63f0b7eb5ebcfb1
5b281df4aaa915f660e075dc944a02c2	add02792cfff7b19b8e526a247acb0ba	ff43f91f2465504e5e67d0b37d92ef18
5e5d9971c90287a6aa905e54b2a21b1c	ae2f3e2412925a767e372c9c0ccf7ced	

Appendix C: POORTRY Certificate Details

The following certificate details are extracted from the certificate signing to the POORTRY sample. However, note that this is a legitimate attestation signing Microsoft certificate. Note that some details were removed for brevity.

```
Certificate:

    Data:

        Version: 3 (0x2)

        Serial Number:

            33:00:00:00:57:ee:4d:65:9a:92:3e:7c:10:00:00:00:00:00:57

        Signature Algorithm: sha256WithRSAEncryption

        Issuer: C = US, ST = Washington, L = Redmond, O = Microsoft Corporation, CN = Microsoft Windows Third Party
Component CA 2014

        Validity

            Not Before: Jun  7 18:08:06 2022 GMT

            Not After : Jun  1 18:08:06 2023 GMT

        Subject: C = US, ST = Washington, L = Redmond, O = Microsoft Corporation, CN = Microsoft Windows Hardware
Compatibility Publisher

        Subject Public Key Info:

            Public Key Algorithm: rsaEncryption

                RSA Public-Key: (2048 bit)

               Exponent: 65537 (0x10001)

        X509v3 extensions:

            X509v3 Extended Key Usage:

                1.3.6.1.4.1.311.10.3.5, 1.3.6.1.4.1.311.10.3.5.1, Code Signing

            X509v3 Subject Key Identifier:

                41:8F:FB:78:B4:1F:1F:7F:19:8E:36:12:08:D0:22:76:6B:58:FA:29

            X509v3 Subject Alternative Name:

                DirName:/OU=Microsoft Operations Puerto Rico/serialNumber=232147+470769

            X509v3 Authority Key Identifier:

                keyid:C8:3A:9C:A7:4A:C3:23:F2:25:7E:B9:DA:AB:29:53:0E:54:00:C3:A1
```