Security Now! #493 - 02-03-15

TOR: Not So Anonymous

This week on Security Now!

- Regin's apparent heritage,
- The "GHOST" vulnerability we talked about last week,
- A clarification about SQRL,
- And the disturbing truth about TOR

Comparing “REGIN” to “QWERTY”

Security News

REGIN made by the FiveEyes team...

- From the "Attribution is Difficult Department"
- Five Eyes: Australia, Canada, New Zealand, the UK & US.
- These countries are bound by the multilateral UKUSA Agreement which is a treaty for joint cooperation in signals intelligence.
- Der Spiegel published some Showden documents detailing NSA malware known internally as QWERTY.

- It ASBOLUTELY shares source code with the keystroke monitor component of REGIN.
- "Researchers Link Regin to Malware Disclosed in Recent Snowden Documents"
- http://threatpost.com/researchers-link-regin-to-malware-disclosed-in-recent-snowden-documents/110667
- Researchers at Kaspersky Lab have discovered shared code and functionality between the Regin malware platform and a similar platform described in a newly disclosed set of Edward Snowden documents 10 days ago by Germany's Der Spiegel.
- The link, found in a keylogger called QWERTY allegedly used by the so-called Five Eyes, leads them to conclude that the developers of each platform are either the same, or work closely together.
- Researchers at Kaspersky Lab wrote: "Considering the extreme complexity of the Regin platform and little chance that it can be duplicated by somebody without having access to its source codes, we conclude the QWERTY malware developers and the Regin developers are the same or working together."
- Snowden: WARRIORPRIDE <--> QWERTY
- http://www.spiegel.de/international/world/regin-malware-unmasked-as-nsa-tool-after-spiegel-publishes-source-code-a-1015255.html
- http://www.theregister.co.uk/2015/01/29/did_aussie_spooks_write_regin/

Bad Linux "GHOST" vulnerability

- Note: The "Openwall" reference was just one appearance of the notification
  - http://www.openwall.com/lists/oss-security/2015/01/27/9

- http://www.zdnet.com/article/critical-linux-security-hole-found/
- GHOST, a critical Linux security hole, is revealed
  Researchers at cloud security company Qualys have discovered a major security hole, GHOST (CVE-2015-0235), in the Linux GNU C Library (glibc). This vulnerability enables hackers to remotely take control of systems without even knowing any system IDs or passwords.

  Qualys alerted the major Linux distributors about the security hole quickly and most have now released patches for it. Josh Bressers, manager of the Red Hat product security team said in an interview that, "Red Hat got word of this about a week ago. Updates to fix GHOST on Red Hat Enterprise Linux (RHEL) 5, 6, and 7 are now available via the Red Hat Network."

  This hole exists in any Linux system that was built with glibc-2.2, which was released on November 10, 2000. Qualys found that the bug had actually been patched with a minor bug fix released on May 21, 2013 between the releases of glibc-2.17 and glibc-2.18.

  However, this fix was not classified as a security problem, and as a result, many stable and long-term-support distributions are wide open today. Linux systems that are liable to attack include Debian 7 (Wheezy), RHEL 5, 6, and 7, CentOS 6 and 7 and Ubuntu 12.04. Besides Red Hat's fix, Debian is currently repairing its core distributions, Ubuntu has patched the bug both for 12.04 and the older 10.04, and I'm told the patches are on their way for CentOS.

The security hole can be triggered by exploiting glibc's gethostbyname functions. This function is used on almost all networked Linux computers when the computer is called on to access another networked computer either by using the /etc/hosts files or, more commonly, by resolving an Internet domain name with Domain Name System (DNS).

To exploit this vulnerability, all an attacker needs to do is trigger a buffer overflow by using an invalid hostname argument to an application that performs a DNS resolution. This vulnerability then enables a remote attacker to execute arbitrary code with the permissions of the user running DNS. In short, once an attacker has exploited GHOST they may be capable of taking over the system.

"GHOST poses a remote code execution risk that makes it incredibly easy for an attacker to exploit a machine. For example, an attacker could send a simple email on a Linux-based system and automatically get complete access to that machine," said Wolfgang Kandek, Qualys's CTO in a statement. "Given the sheer number of systems based on glibc, we believe this is a high severity vulnerability and should be addressed immediately. The best course of action to mitigate the risk is to apply a patch from your Linux vendor."

Unlike some security announcements, Kandek is not crying wolf. Qualys has developed a proof-of-concept in which simply sending a specially created e-mail to a mail server enabled them to create a remote shell to the Linux machine. According to Qualys, "This bypasses all existing protections (like ASLR, PIE and NX) on both 32-bit and 64-bit systems."

- Qualys Security Advisory CVE-2015-0235 - GHOST: glibc gethostbyname buffer overflow

  During a code audit performed internally at Qualys, we discovered a buffer overflow in the __nss_hostname_digits_dots() function of the GNU C Library (glibc). This bug is reachable both locally and remotely via the gethostbyname*() functions, so we decided to analyze it -- and its impact -- thoroughly, and named this vulnerability "GHOST".

  Our main conclusions are:

  - - Via gethostbyname() or gethostbyname2(), the overflowed buffer is located in the heap. Via gethostbyname_r() or gethostbyname2_r(), the overflowed buffer is caller-supplied (and may therefore be located in the heap, stack, .data, .bss, etc; however, we have seen no such call in practice).

  - At most sizeof(char *) bytes can be overwritten (ie, 4 bytes on 32-bit machines, and 8 bytes on 64-bit machines). Bytes can be overwritten only with digits ('0'...'9'), dots ('.'), and a terminating null character ('\0').

  - Despite these limitations, arbitrary code execution can be achieved. As a proof of concept, we developed a full-fledged remote exploit against the Exim mail server, bypassing all existing protections (ASLR, PIE, and NX) on both 32-bit and 64-bit

machines. We will publish our exploit as a Metasploit module in the near future.
    - The first vulnerable version of the GNU C Library is glibc-2.2, released on November 10, 2000.

    - We identified a number of factors that mitigate the impact of this bug. In particular, we discovered that it was fixed on May 21, 2013 (between the releases of glibc-2.17 and glibc-2.18). Unfortunately, it was not recognized as a security threat; as a result, most stable and long-term-support distributions were left exposed (and still are): Debian 7 (wheezy), Red Hat Enterprise Linux 6 & 7, CentOS 6 & 7, Ubuntu 12.04, for example.

- http://threatpost.com/php-applications-wordpress-subject-to-ghost-glibc-vulnerability/110755
    - PHP code and Wordpress are likely vulnerable.

SQRL

Jay Littlefield in San Francisco
Subject: SQRL question regarding compromised web sites
Date: 10 Jan 2015 23:25:26
:
Hello Steve and Leo!

Fan of the show here. I've been listening to Security Now on my commute for the past several years, am a proud owner of SpinRite, and, thanks to you two, a Harry's razor convert. I really appreciate the great shows you produce. I'm also very excited about SQRL and hope to use it on my own web sites as it becomes available within the major website development packages.

Steve, I have a SQRL question for you that I have not been able to find the answer to in any of your podcasts or on your website. You often cite the fact that SQRL creates unique public / private key pairs for each individual website accessed. Because of this, a breach of one website will not compromise your identity on any other website, unlike the common practice of reusing passwords. This is a great improvement, but what about your identity on the compromised site?

Let's say I am an active SQRL user for all of my web transactions. I read about a major security breach at say, Target.com, where I am an active SQRL user. Let's assume (hypothetically) that their entire customer database has been compromised, and I am instructed to reset my password for my account. If I'm using a password, I can do that. But passwords are supposed to be archaic once SQRL arrives. If so, then what's the SQRL equivalent of a password reset for an individual site in the event of a breach?

You've mention SQRL has the ability to change your Master ID should **IT** become compromised, but a breach of my SQRL credentials at Target.com, by definition, does not compromise my identity anywhere else on the web. Can you elaborate on how this situation should be handled by a SQRL user? I'm afraid the answer to this question is currently lost on me.

Thanks again for a wonderful show and all you do for your community of followers! Regards,
-- Jay

SpinRite

Igor Koveshnikov in New Jersey
Subject: Testimonial/Request/Question
Date: 20 Jan 2015 06:16:26

Hello Steve and Leo!

About a week ago my former boss brought me his laptop that wouldn't boot. I remember I installed a Crucial M4 256 GB SSD in it, and was puzzled what could be wrong with SSD. When I turned on the laptop I could get to first mouse cursor appearance in Windows and then HD light would become solid lit and nothing would happen. I connected the SSD using "USB to SATA" adapter to my computer. I could read the drive, but it took me about 4 days to retrieve 170 GB of documents. It would freeze on some files for long minutes and then would continue to copy. It behaved exactly like damaged mechanical hard drive, just without a clicking noise. Later it turned out that my boss followed my suggestion and all his documents were backed up by Carbonite. Still I always try to retrieve most of data before I started playing with drives.

I always wanted to try Spinrite on SSD, but it doesn't make sense to run maintenance on them. Here was my opportunity and I used it. Started on level 1 just to see what Spinrite makes of the drive. It run quickly, showed some "R"s, but nothing changed when I tested a boot. Level 2, however, completely fixed the drive. When I rebooted the laptop, I got to login screen and the laptop runs like new.

It's easy to understand what happens to damaged mechanical drives, but really hard when it comes to SSDs. My suspicion it's software/firmware issue, something happens to internal table of cell assignments, but then how Spinrite is able to fix it?..

TOR: Not so Anonymous after all

Our previous coverage:

- SN#70 (Internet Anonymity) - seven years ago, March 28th, 2008
- SN#394 (TOR Hidden Services) - nearly two years ago, March 8th, 2013
- In our earlier "what is TOR" coverage, we primarily focused upon the cleverness of TOR's ONION layering cryptography.

http://thestack.com/chakravarty-tor-traffic-analysis-141114

- "81% of Tor users can be de-anonymised by analysing router information, research indicates."
- Using weak but pervasive built-in Cisco "NetFlow" tech and deliberate traffic perturbation.

- Perturb the traffic from the server a user is connecting to, and watch the exit nodes' traffic.
- The point was that even very weak "NetFlow" aggregation was enough. More expensive "per packet" monitoring and analysis was not needed.

Did feds mount a sustained attack on Tor to decloak crime suspects?

- http://arstechnica.com/tech-policy/2015/01/did-feds-mount-a-sustained-attack-on-tor-to-decloak-crime-suspects/
- <quote> Despite the use of Tor, FBI investigators were able to identify IP addresses that allegedly hosted and accessed the servers, including the Comcast-provided IP address of one Brian Farrell, who prosecutors said helped manage SilkRoad2. In the affidavit, DHS special agent Michael Larson wrote:
    - From January 2014 to July 2014, a FBI NY Source of Information (SOI) provided reliable IP addresses for TOR and hidden services such as SilkRoad2, which included its main marketplace URL, its vendor URL, its forum URL, and its support interface (uz434sei7arqunp6.onion). The SOI's information ultimately led to the identification of SilkRoad2 servers, which led to the identification of at least another seventeen black markets on TOR.
    - The SOI also identified approximately 78 IP addresses that accessed a vendor .onion address. A user cannot accidentally end up on the vendor site. The site is for vendors only, and access is only given to the site by the SilkRoad2 administrators/moderators after confirmation of a significant number of successful transactions. If a user visits the vendor URL, he or she is asked for a user name and password. Without a user name and password, the vendor website cannot be viewed.

The Internet was never designed to provide anonymity... and it doesn't.

- True anonymity is extremely difficult to achieve.
- In a high-latency store & forward system it's somewhat feasible...
- But in any low-latency near real time network, it's arguably impossible.

Review... What is TOR?

- TOR is a LOW LATENCY anonymity-enhancing network service.
- The original designers of TOR made some assumptions and compromises that are coming back to haunt us now...
- One academic paper put it this way: "Tor aims to protect against a peculiar threat model, that is unusual within the anonymous communications community. It is conventional to attempt to guarantee the anonymity of users against a global passive adversary, who has the ability to observe all network links. It is also customary to assume that transiting network messages can be injected, deleted or modified and that the attacker controls a subset of the network nodes. This models a very powerful adversary, and systems that protect against it can be assumed to be secure in a very wide range of real world conditions.

  Tor, on the other hand, assumes a much weaker threat model. It protects against a (weaker) non-global adversary, who can only observe a fraction of the network, modify the traffic only on this fraction, and control a fraction of the Tor nodes.

  Furthermore, Tor does not attempt to protect against traffic confirmation attacks, where

an adversary observes two parties that he suspects to be communicating with each other, to either confirm or reject this suspicion. Instead, Tor aims to make it difficult for an adversary with a very poor a priori suspicion of who is communicating with whom, to gain more information.

The Crypto Model:

- Choose a "circuit", default is three nodes.
- Negotiate keys with the 1st node.
- Using the first node, get keys for a randomly chosen second node.
- Using the first and second nodes, get keys for the randomly chosen third node.
- Wrap outgoing traffic in an onion from node 3 to node 2 to node 1.
- The onion model nailed it. No one is attacking that. But...

The Traffic Flow Model: (and the Achilles' heel)

- Deliberate obfuscation of individual packets with random length padding.
- TCP flows are divided into 512 byte cells... And are sent round robin out of the node.
- The power of the global observer
- Much like metadata... traffic pattern analysis is a POWERFUL tool.
- The power of active vs passive attacks
- Being able to "perturb" the flow makes attacks far more powerful.

The extreme power of active assumption confirmation attacks.

- One academic paper: <quote> "Tor does not attempt to protect against traffic confirmation attacks, where an adversary observes two parties that he suspects to be communicating with each other, to either confirm or reject this suspicion."
- IOW -- In any near real time network, traffic confirmation is a killer.

Bottom line... *I* would never rely upon TOR alone.

- Consider it, itself, another layer of a more full "Defense in Depth."
- The dream is that someone can sit at home and be fully anonymous. But that's not the reality.

Defense in depth:

- First of all... DO NOT do anything illegal. Do not do anything that you wouldn't want the Federal Government to know about.
- Traditional old school & new school.
- Go somewhere as far away as convenient.
- Be anonymous there... Pay with cash.
- Don't go anywhere familiar, don't stay long, don't know anyone, don't talk to anyone.
- Plan ahead to get in and out. Rehearse for speed. Get it done and leave.
- Don't do ANYTHING having to do with your own identity.
- Perhaps purchase a cheap laptop just for this. Pay with cash.
- Override your laptop's default MAC address.
- Use TOR and sacrifice real time performance
- Use widely dispersed global nodes.
- Use many nodes.
-
- In other words... Tor IS useful, but it's not perfect. So always act as though it's not.