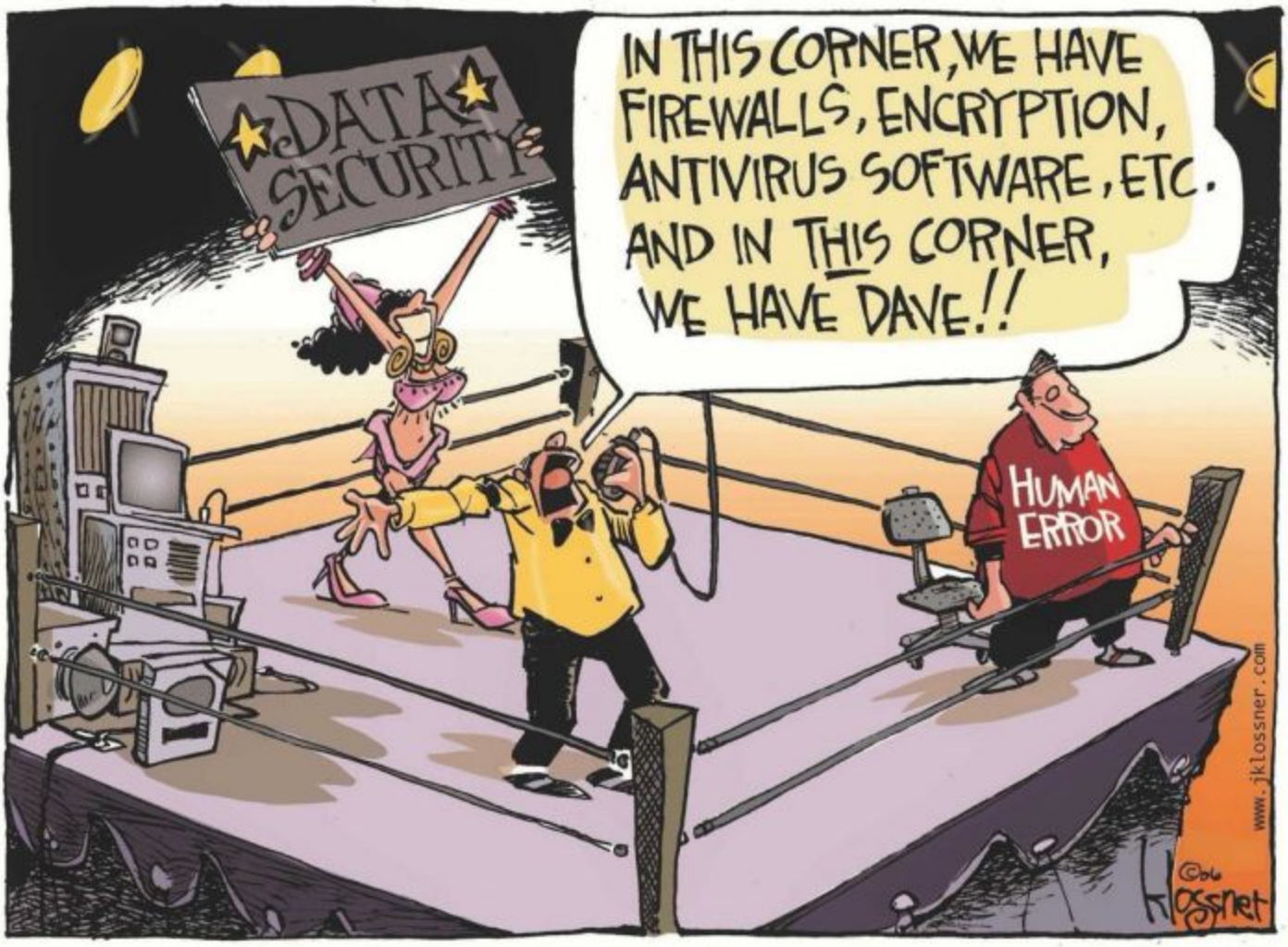
Security Now! #618 - 06-27-17

Research: Useful and Otherwise

This week on Security Now!

This week we discuss another terrific NIST initiative, RSA crypto in a quantum computing world, Cisco's specious malware detection claims, the meaning of post-audit OpenVPN bug findings, worrisome bugs revealed in Intel's recent Skylake and KabyLake processors, the commercialization of a malware technique, WannyCry keeps resurfacing, LinkSys responds to the CIA's Vault7 CherryBomb firmware, another government reacts to encryption, the NSA's amazing Github repository, more news about HP printer auto-updating, a piece of errata, some miscellany, and some closing the loop feedback from our listeners.

Our Picture of the Week



Security News

NIST's Information Technology Laboratory (ITL)
Introduction

The world is increasingly interconnected by myriads of devices, working in concert to accomplish important tasks, including automotive systems, sensor networks, healthcare, distributed control systems, the Internet of Things (IoT), cyber-physical systems, and the smart grid. But these devices may have resource restrictions, compared to common desktop computers. For example, they may have significantly reduced power consumption, less computation power, and orders of magnitude less memory than desktop computers. These constraints can make it difficult to implement modern cryptographic algorithms, most of which are designed for desktop or server environments. To address this issue, different cryptographic algorithms have been tailored for resource-constrained devices. The academic community has performed a significant amount of work on this type of cryptography, called lightweight cryptography. This work includes efficient implementations of conventional cryptography standards and the design and analysis of new lightweight algorithms and protocols.

In 2013, NIST's Information Technology Laboratory started a lightweight cryptography project to investigate the issues and then develop a strategy for the standardization of lightweight cryptographic algorithms. In 2015 and 2016, NIST held two lightweight cryptography workshops to solicit public feedback on the constraints and limitations of the target devices, and the requirements and characteristics of real-world applications of lightweight cryptography.

Recently, NIST decided to create, through an open process, a portfolio of lightweight algorithms. ITL has published NIST Internal Report (NISTIR) 8114, Report on Lightweight Cryptography, to summarize the findings of this project and to outline NIST's plans for the standardization of lightweight algorithms. Here, we present highlights from this report, especially the devices targeted by lightweight cryptography, how the algorithms were designed, and the standardization of lightweight cryptographic algorithms.

Report on Lightweight Cryptography
http://nvlpubs.nist.gov/nistpubs/ir/2017/NIST.IR.8114.pdf

This report provides an overview of lightweight cryptography, summarizes the findings of NIST's lightweight cryptography project, and outlines NIST's plans for the standardization of lightweight algorithms. In particular, NIST has decided to create a portfolio of lightweight algorithms through an open process. This report includes a list of questions to the stakeholders of lightweight cryptography that will serve as the basis for determining requirements. NIST will develop profiles based on community responses to these questions. These profiles are intended to capture cryptographic algorithm requirements imposed by devices and applications where lightweight cryptography is needed. Algorithms will be recommended for use only in the context of profiles, which describe physical, performance, and security characteristics.

Table of contents:

For example, under Lightweight Block Ciphers:

- Smaller Block Size:
  To save memory, lightweight block ciphers may use smaller block sizes than AES (e.g., 64 bits or 80 bits, rather than 128 bits). It should also be noted that using small block sizes reduces limits on the maximum number of plaintext blocks to be encrypted. For example, outputs of a 64-bit block cipher can be distinguished from a random sequence using around 2^32 blocks for some of the approved modes of operations. Depending on the algorithm, this may lead to attacks such as plaintext recovery or key recovery or with non-negligible probabilities.

- Smaller Key Sizes:
  Some lightweight block ciphers use small key sizes (less than 96 bits) for efficiency (e.g., 80-bit PRESENT). At the time of this writing, the minimum key size required by NIST is 112 bits

- Fewer Rounds:
  The components and operations used in lightweight block ciphers are typically simpler than those of conventional block ciphers. In lightweight designs using S-boxes, 4-bit S-boxes are preferred over 8-bit S-boxes. This reduction in size results in significant area savings. For example, the 4-bit S-box used in PRESENT required a relative area of 28, whereas the AES S-box required 395. For hardware-oriented designs, bit permutations (such as those used in PRESENT), or recursive MDS matrices (as in PHOTON [24] and LED [25]) may be preferred over complex linear layers. When rounds are simpler, they may need to be iterated more times to achieve security.

- Simpler key schedules:
  Complex key schedules increase the memory, latency and the power consumption of implementations; therefore, most of the lightweight block ciphers use simple key schedules that can generate sub-keys on the fly. This may enable attacks using related keys, weak keys, known keys or even chosen keys. Using a secure key derivation function (KDF) can prevent some of these attacks.

- Minimal implementations:
  There are several modes of operation and protocols that require only the encryption function of a block cipher. Some applications may require a device to only support one of the encryption or decryption operations. Implementing only the necessary functions of a cipher may require fewer resources than implementing the full cipher.

This is all for the best, and I welcome this sort of standards effort. It's similar to the nearly 4-year long NIST-sponsored competition that resulted in Rijndael becoming the AES standard cipher and the following competition for the SHA-3 standard which was won by the Keccak hash.

So... While having a suite of carefully considered lightweight cryptographic primitive building blocks does not automatically make IoT devices secure, it falls into the classification of "necessary but not sufficient." As such, it removes one of the largest objections that IoT makers can present: Crypto raises the cost and makes secure device non-competitive. We already know that everyone WANTS crypto, but no one wants to actually PAY for it. So this helps to zero the effective cost of tomorrow's secure devices.

Post-quantum RSA / Dan Bernstein and three others...
http://eprint.iacr.org/2017/351.pdf

Abstract:
This paper proposes RSA parameters for which key generation, encryption, decryption, signing, and verification are feasible on today's computers while all known attacks are infeasible, even assuming highly scalable quantum computers. As part of the performance analysis, this paper introduces a new algorithm to generate a batch of primes. As part of the attack analysis, this paper introduces a new quantum factorization algorithm that is often much faster than Shor's algorithm and much faster than pre-quantum factorization algorithms. Initial pqRSA implementation results are provided.

Shor's Algorithm: Named after the mathematician Peter Shor who, 23 years ago, back in 1994, proposed a quantum algorithm (assuming future quantum computers) which is able to perform high-speed integer factorization.

As we know, it has been the longstanding intractability of the integer factorization problem -- that is, the difficulty of determining the two prime number factors which comprise a public key -- which is the basis for RSA asymmetric key technology. In RSA, the public key IS the product of two secret prime factors, and the reason a public key can be made public is that even knowing it, we do not currently know how to break it back apart into the two prime factors that were multiplied to obtain to it in the first place.

But with factorization, this is a place where size definitely matters. We all know the prime factors of '35'. That's not difficult. But it's easy because there are not many possible prime factors smaller than 35. This is not true when the products of primes are several thousands of bits long. We know that primes are surprisingly common and that they do not get less common as numbers grow large. So there are too many possibilities, and no one has found a highly efficient way on non-quantum computers to break apart a sufficiently lengthy product of two larger primes.

The essential question this paper seeks to answer is, even assuming the presence of fast quantum factoring (and while offering a new faster quantum factorization algorithm called GEECM), under quantum factorization, how does the difficulty of usage scale relative to the difficult of practical attack as the prime factor product length increases?

They ask: Is it actually true that quantum computers will kill RSA?

And they write: The question here is not whether quantum computers will be built, or will be affordable for attackers. This paper assumes that astonishingly scalable quantum computers will be built, making a qubit operation (a quantum bit) as inexpensive as a bit operation. Under this assumption, Shor's algorithm easily breaks RSA as used on the Internet today. The question is whether RSA parameters can be adjusted so that all known quantum attack algorithms are infeasible while encryption and decryption remain feasible.

The conventional wisdom is that Shor's algorithm factors an RSA public key 'n' almost as quickly as the legitimate RSA user can decrypt. Decryption uses an exponentiation modulo n; Shor's algorithm uses a quantum exponentiation modulo n. There are some small overheads in Shor's algorithm, but these overheads create only a very small gap between the cost of decryption and the cost of factorization. (Shor speculated that faster quantum algorithms for modular exponentiation could even make breaking RSA on a quantum computer asymptotically faster than encrypting with RSA on a classical computer"; however, no such algorithms have been found.)

The main finding of this paper is that standard techniques for speeding up RSA, when pushed to their extremes, create a much larger gap between the legitimate user's costs and the attacker's costs. Specifically, for this paper's version of RSA, the attack cost is essentially quadratic in the usage cost.

These extremes require a careful analysis of quantum algorithms for integer factorization. As part of this security analysis, this paper introduces a new quantum factorization algorithm, GEECM, that is often much faster than Shor's algorithm and all pre-quantum factorization algorithms. See Section 2. GEECM turns out to be one of the main constraints upon parameter selection for postquantum RSA.

These extremes also require a careful analysis of algorithms for the basic RSA operations. As part of this performance analysis, this paper introduces a new algorithm to generate a large batch of independent uniform random primes more efficiently than any known algorithm to generate such primes one at a time.

And finally, we report initial implementation results for RSA parameters large enough to push all known quantum attacks above 2^100 qubit operations. These results include successful completion of the most expensive operation in post-quantum RSA, namely ... the generation of an 8-terabit public key.

In other words, this is interesting core crypto research that often serves to point the way to further work, and from which other researchers will doubtless draw inspiration.

No one is suggesting that we should actually keep using RSA in a post-quantum world and make it quantum-proof by schlepping around 8-terabit public keys. Yeah... we could, but we're not going to. There are already a bunch of next-generation quantum-hardened crypto systems under development.

Cisco is claiming to be able to detect malware within encrypted web traffic with 99$ accuracy.

Last October, at the "AISec 2016 Proceedings of the 2016 ACM Workshop on Artificial Intelligence and Security, Cisco's Blake Anderson and David McGrew delivered their research paper: "Identifying Encrypted Malware Traffic with Contextual Flow Data"

> Abstract:
> Identifying threats contained within encrypted network traffic poses a unique set of challenges. It is important to monitor this traffic for threats and malware, but do so in a way that maintains the integrity of the encryption. Because pattern matching cannot operate on encrypted data, previous approaches have leveraged observable metadata gathered from the flow, e.g., the flow's packet lengths and inter-arrival times.
>
> [Remember that we have previously encountered and reported on numerous side-channel attacks on encryption which leveraged content-dependent compression ratios, which are preserved by and can be observed through compression, to infer the pre-encrypted content.]
>
> In this work, we extend the current state-of-the-art by considering a "data omnia" approach. To this end, we develop supervised machine learning models that take advantage of a unique and diverse set of network flow data features. These data features include TLS handshake metadata, DNS contextual flows linked to the encrypted flow, and the HTTP headers of HTTP contextual flows from the same source IP address within a 5 minute window.
>
> We begin by exhibiting the differences between malicious and benign traffic's use of TLS, DNS, and HTTP on millions of unique flows. This study is used to design the feature sets that have the most discriminatory power. We then show that incorporating this contextual information into a supervised learning system significantly increases performance at a 0.00% false discovery rate for the problem of classifying encrypted, malicious flows. We further validate our false positive rate on an independent, real-world dataset.

Uhhhhh... so they have a 0.00% false positive discovery rate. Okay. But so does a can of dog food. You can set any can of dog food down in the middle of a desk... and you can even plug it in if you want to. It doesn't matter either way. And, in fact, if you DON'T plug it in, it becomes an extremely energy efficient malware detector with a 0.00% false positive malware discovery rate. No matter what happens, it will absolutely never generate a falsely positive warning of malware.

My point is that their paper's Abstract is suspiciously silent on the topic of their false NEGATIVE detection rate -- i.e. how much malware whizzes past that they miss.

I chose not to waste the $15 required to obtain the full text of their paper. But I already had a related paper they published earlier last summer titled "Deciphering Malware's use of TLS (without Decryption)." In that paper they look at TLS handshake metadata and conclude that after training on the TLS handshake characteristics of various families of known malware, they are able to determine, with 90% accuracy, which family of known malware the subsequent connection carried after the handshake.

But, of course, discriminating among the malware family of known malicious connections is an ENTIRELY different problem from spotting malware among a flood of benign traffic.

Research is fine, and I'm not meaning to suggest that metadata analysis is not at all useful for some purposes. But last week, Cisco said it has developed technology capable of spotting malware inside secure data without having to decrypt the traffic. That, Cisco says, means corporate customers don't have to choose between privacy and security.

Cisco's senior VP David Goeckeler said at a press event in San Francisco, Tuesday: "We get both privacy and security." He added that the new technology can detect malware with 99% accuracy.

Cisco plans to offer the encrypted traffic analytics to customers as a subscription service, part of a growing effort to build that side of its business.

My advice?? This has every earmark of being a low-success-rate heuristic which is attempting to do something sexy and desirable... and impossible. So don't reply on it exclusively at first, and test it thoroughly in a network lab setting before signing any long-term contract.

So the unfortunate title of this work is: "The OpenVPN post-audit bug bonanza"
(Fuzzing can, indeed, discover otherwise hidden bugs.)
https://guidovranken.wordpress.com/2017/06/21/the-openvpn-post-audit-bug-bonanza/
Guido Vranken

> Summary
> [He writes] I've discovered 4 important security vulnerabilities in OpenVPN. Interestingly, these were not found by the two recently completed audits of OpenVPN code. Below you'll find mostly technical information about the vulnerabilities and about how I found them, but also some commentary on why commissioning code audits isn't always the best way to find vulnerabilities.
>
> After a hardening of the OpenVPN code (as commissioned by the Dutch intelligence service AIVD) and two recent audits 1 2, I thought it was now time for some real action ;).
>
> Most of this issues were found through fuzzing. I hate admitting it, but my chops in the arcane art of reviewing code manually, acquired through grueling practice, are dwarfed by the fuzzer in one fell swoop; the mortal's mind can only retain and comprehend so much information at a time, and for programs that perform long cycles of complex, deeply nested operations it is simply not feasible to expect a human to perform an encompassing and reliable verification.
>
> End users and companies who want to invest in validating the security of an application written in an "unsafe" language like C, such as those who crowd-funded the OpenVPN audit, should not request a manual source code audit, but rather task the experts with the goal of ensuring intended operation and finding vulnerabilities, using that strategy that

provides the optimal yield for a given funding window.

Upon first thought you'd assume both endeavors boil down to the same thing, but my fuzzing-based strategy is evidently more effective. What's more, once a set of fuzzers has been written, these can be integrated into a continuous integration environment for permanent protection henceforth, whereas a code review only provides a "snapshot" security assessment of a particular software version.

Manual reviews may still be part of the effort, but only there where automation (fuzzing) is not adequate.

I strongly disagree that EITHER approach is more or less valuable than the other. The two approaches are different, and it that difference that makes BOTH important. There are definitely many classes of critical problems that fuzzing won't see. For example, subtle flaws in cryptographic implementations, secrets-based timing and power changes that would enable side-channel attacks, unsafe assumptions about the use of fundmental cryptographic primitives... and I could go on all day. So this notion that deliberate code auditing is not every bit as important as fuzzing is nonsense.

That said... as we've discussed here many times, fuzzing is a very powerful tool for discovering an entirely DIFFERENT class of also very important bugs that could, as Guido correctly asserts, easily slip past code auditors. Fuzzing is super-cheap and super-easy once it's been setup. So, yes... I'm delighted that he did this and that as a result the world now has an even stronger OpenVPN system than it had before. But no one should think for a second that any money could have been saved by fuzzing INSTEAD of careful code auditing.

But Guido's important work demonstrates that BOTH approaches should always be used.

Intel's Skylake and KabyLake processors have been found to contain serious microcode bugs

(These problems will be resolved, but I'm happy not to have them as a coincidence of having jumped onto Haswell processors for my next workstation and laptop... even though Microsoft later reversed themselves on their plans not to fully support Windows 7 on older chips.)

As we know, low level processor flaws are not without precedent, but fortunately they are quite rare and our chip vendors generally do an amazing job with the stunning complexity of modern CISC (complex instruction set) processors.

So far, Intel has been relatively quiet about the problem, but engineers at Dell and Intel have told reporters that the problem, and its fix, exists. The microcode patch is currently being quietly tested to make very sure that it doesn't break anything else.
Naturally, being a processor-level bug, all operating systems and other software running on broken chips -- Windows, macOS, Linux, FreeBSD, etc. and their applications -- can be vulnerable.

Quietly-published Intel chip errata reads: "Under complex micro-architectural conditions, short loops of less than 64 instructions that use AH, BH, CH or DH registers as well as their

corresponding wider register (e.g. RAX, EAX or AX for AH) may cause unpredictable system behavior. This can only happen when both logical processors on the same physical processor are active."

"Both logical processors" means hyperthreading. This is a subtle hyperthreading problem. The problem was first uncovered back in January by developers working on the Ocaml language compiler when things didn't seem to be working right and the problem didn't appear to be their code. It was traced back to Intel processors that appeared on the market a little more than a year ago.

Regardless of which OS you're using, the Debian page have the most complete information

[WARNING] Intel Skylake/Kaby Lake processors: broken hyper-threading

- https://lists.debian.org/debian-devel/2017/06/msg00308.html
- http://bit.ly/sn618

If you want to be proactive, disabling hyperthreading in your system's BIOS will prevent the trouble until chip firmware updates are more widely available. Since these are today's chips, I'm sure that all of the major OS vendors will soon be providing processor firmware updates where possible and appropriate.

Before You Hit 'Submit,' This Company Has Already Logged Your Personal Data
http://gizmodo.com/before-you-hit-submit-this-company-has-already-logge-1795906081

So you visit a website and begin filling in a contact form or their "create an account" form, or perhaps purchase information... or perhaps your handy-dandy browser, which has been paying attention, beats you to it on some fields, auto-filling a bunch of the page's form fields for you.

In the traditional world we all grew up in, this was all a passive process. Somewhere on the page, typically at the bottom below the fill-in area, is a "Create an account" or a "Submit" button... and the implicit and longstanding rule has been... nothing happens with any of the information until and unless you press that button to explicitly and deliberately send that provided information off to the hosting web server.

A few months back we talked about malicious page code that was deliberately creating unseen offscreen form fields which the browser's auto-fill automation would fill in with well-meaning intent. And just that would be sufficient for malicious page code to suck the browser-filled and unseen field contents off the page and spirit them away to parts unknown.

On the heels of the adage "if it can be done someone will do it" we now seeing that "if someone can make money doing it, they will." And in this case an ultra slimy company named NaviStone...

https://www.navistone.com/

From their public website:

- "Reach your anonymous website visitors with Retargeting Postcards and convert 50X more visitors than digital display ads."
- "Convert your anonymous website traffic"
- "Harness the power of consumer intent data."
- Take Your Retargeting Offline.
- Until today, your ability to retarget anonymous website shoppers was limited to low-impact, low-response digital display ads.
- With the NaviStone® turn-key Postcard program, you can:
    - Reach out directly to anonymous website shoppers.
    - Tailor your marketing to the individual shopper based on their website behavior.
    - Mail personalized postcards within 24-48 hours after a site visit.

Who we are:

> Traditional direct marketing contact strategies are driven entirely by past purchase behavior. As that data ages, it becomes less predictive of future responsiveness. At NaviStone®, we lead the vanguard in progressive website visitor tracking technology. Our proven approach to both customer acquisition and reactivation allows us to create a list of unique, engaged website visitors to include in your marketing campaigns.
>
> Our technology adds the power of intent — digital browsing behavior, to traditional transaction history to expand the scope and improve the productivity of your direct and digital marketing programs.

So how does this relate to filling out forms on the web?

Sadly, Navistone is reportedly selling their ability to harvest your UNSUBMITTED (and possibly automatically filled-in) form data the moment you visit a site's webpage that uses this technology. And this data does not go to the hosting site. No. It goes back to servers controlled by Navistone where it is doubtless, if we believe the claims on their website, merged into existing tracking-derived data and used to compile marketing information which is then sold back to their clients.

During Gizmodo's testing, they write "Three sites—hardware site Rockler.com, gift site CollectionsEtc.com, and clothing site BostonProper.com—sent us emails about items we'd left in our shopping carts using the email addresses we'd typed onto the site but had never submitted." A search using the "BuiltWith" service which shows the technology used by websites revealed more than 100 sites using this NaviStone technology.

This capability is enabled by the full standardization of a scriptable DOM -- Document Object Model -- in all modern web browsers. This allows script running on the page to completely traverse the entire page hierarchy, examining, reading and writing any and all page content and

components. It is an EXTREMELY powerful capability and can so easily be abused.

Unfortunately, when even a well-meaning website allows any third-party such as NaviStone to invoke their own JavaScript onto their site's pages, all visitors to that site are inherently trusting all of the actions of that 3rd party. Perhaps the most worrisome possibility would be that a web browser or password manager might auto-fill your username and password for a site you visit and that a company like NaviStone would suck up that data the moment it was filled in.

Gizmodo writes: NaviStone is an Ohio-based startup in the business of identifying "ready to engage" customers and matching "previously anonymous website visitors to postal names and addresses." It says it can send postcards to the homes of anonymous website shoppers within a day or two of their visit, and that it's capable of matching "60-70% of your anonymous site traffic to Postal names and addresses."

Troubling as this is, we need to remember that this capability, while perhaps more troubling and worrisome now, is not completely new. Our longest listening podcast followers will recall the true experience I shared of one of our earliest listeners in Toronto Canada. He was planning a visit South to New York, so he went to the New York Symphony's website to see what was on their upcoming calendar. He entered nothing and purchased nothing... just passively browsed. He then shut off the computer and went outside to do some gardening. Not long after the phone rang... and it creeped him out: It was someone representing the New York Symphony saying that they had noticed that he had recently been exploring their schedule and wondered whether there was anything he might wish to pursue. As a listener to this podcast he was more than a bit put off by the tracking that this event had to have revealed.

So... a heads-up to our listeners. As we have discussed in the past, widespread blocking of JavaScript has become progressively less practical as websites increasingly rely upon client-side code running in user's browsers. I would LOVE to be able to block my browser's access to NaviStone's servers so that their script could never run on my browsers. But, unfortunately, they haven't made that easy either. Gizmodo reports that scrapped browser page data is being sent to servers at the "murdoog.com" domain. Murdoog's ICANN registration shows murdoog.com registered by a Tom White of the "Melasa Group, LLC" but Tom is also shown as NaviStone's Chief Technology Officer.

That this is occurring on a user's browser, without their knowledge or permission, when they visit some other site, for the purpose of harvesting any non-submitted and possibly inadvertently supplied data, as well as for the explicit purpose of deanonymizing a site's anonymous visitor traffic... so that they can them be sent postcards in the mail, represents a troubling abuse of the hosting website's visitor trust.

WannaCry continues to ferret out and infect more systems

- WannaCry Forces Honda to Take Production Plant Offline
- https://www.darkreading.com/attacks-breaches/wannacry-forces-honda-to-take-production-plant-offline-/d/d-id/1329192

The engine and vehicle assembly lines at Honda's Sayama Auto Plant in Japan was stopping for nearly 48 hours last week when a number of systems critical to the assembly line's operation were hit by the WanaCry worm.

And this wasn't the first time: A month earlier Honda believed that it had mitigated and dealt with an initial WannaCry infection.

Reuters and other outlets noted that systems at multiple Honda plants in Asia, North America, Europe, and China were found similarly infected with WannaCry.

- WannaCry Ransomware Infects 55 Speed and Red-Light Cameras in Australia
- https://www.bleepingcomputer.com/news/security/wannacry-ransomware-infects-55-speed-and-red-light-cameras-in-australia/

Meanwhile... some are celebrating the fact that WannaCry has also taken 55 Australian automated red-light violation cameras offline. The infection apparently occurred during maintenance operations when a technician connected an infected USB to the devices, which were apparently running on a Windows OS.

The moral for us is: If you run your stoplight cameras (or your nuclear submarines) on a consumer operating system such as Windows, you get exactly what you pay for.

LinkSys responds to the CherryBlossom CIA/Vault7 attack
http://www.linksys.com/us/support-article?articleNum=263800
Advisory Date: 6/21/17

Overview

Linksys is aware of the CherryBlossom project that was recently released by WikiLeaks' Vault 7 publication. Based on the WikiLeaks report customized firmware was created for certain older Linksys routers without our knowledge or consent for the purposes of monitoring, controlling, and manipulating internet traffic of a "targeted" user.

This customized firmware can be loaded onto a router using one of the following methods:

- physical access to the router
- proximity access to the router via Wi-Fi
- intercepting the device in transit to be delivered to a user

Solution
If users believe their router firmware may have been compromised, Linksys recommends that users download the latest available firmware from http://www.linksys.com/support/ and update your router.

After the update, please perform a factory reset to ensure no remnants of the compromise remain. Instructions on how to do a factory reset can be found here. If users are not able to perform a firmware update or receive an error message during the update, please contact customer support for further instructions.

We would also like to recommend the following changes after the factory reset is complete to further secure the router:

- Set a strong admin password (one that includes capital letters, numbers, special characters, and a password length of at least 8 characters)
- Disable Guest Access if it is not in use
- Disable router features (like WPS ***and UPnP***) if they are not being used

Last week's episode was titled "When Governments React"

- We looked at France, Britain, Japan, Germany & Russia
- Meanwhile, we have reporting that Australia intends to push for "Encryption Backdoors" at the next "Five Eyes" meeting.

- The "Five Eyes" alliance comprises Australia, Canada, New Zealand, the UK and the US. These countries are bound by the multilateral UKUSA Agreement for joint cooperation in signals intelligence, military intelligence, and human intelligence.

- It's always important for us to remember that when any non-technical policy people use terms like "Encryption Backdoors" WE have absolutely no idea what they mean... because neither do THEY. So, until we see actual legislation, all we can infer from this is that they're not happy.

- So, in this case, Australian Attorney General George Brandis is stating he'll be pushing for backdoors at the upcoming meeting of the Five Eyes in Ottawa, Canada next week where they will discuss tactics to combat terrorism and protect borders. Australia has made it clear it wants tech companies to do more than to give intelligence and law enforcement agencies access to encrypted communications.

- Brandis said in a joint statement: “I will raise the need to address ongoing challenges posed by terrorists and criminals using encryption. These discussions will focus on the need to cooperate with service providers to ensure reasonable assistance is provided to law enforcement and security agencies.”

- Brandis has previously rationalized away potential objections to backdooring encryption, reasoning that people's tendency to overshare on social media indicates they won't care if the government (or several governments, actually) has access to their private messages.

It turns out, the NSA has an amazing Github repository!
https://nationalsecurityagency.github.io/

- Apache Accumulo
  A sorted, distributed key/value store that provides robust, scalable data storage and retrieval. It adds cell-based access control and a server-side programming mechanism that can modify key/value pairs at various points in the data management process.

- CASA
  Identifies unexpected and prohibited Certificate Authority certificates on Windows systems.

- CONTROL FLOW INTEGRITY RESEARCH
  A proposed hardware-based method for stopping known memory corruption exploitation techniques described in the "Hardware Control Flow Integrity for an IT Ecosystem" research paper.

- DCP
  A program that reduces the timespan needed for making a forensic copy of hard drives for forensic analysis.

- EOWS
  A web enabled prototype tool that implements the Open Checklist Interactive Language (OCIL) capabilities for creating, managing, and responding to questionnaires.

- FEMTO
  An indexing and search system for queries on sequences of bytes that offers lightning-fast searches on data of arbitrary formats.

- GOSECURE
  An easy to use and portable Virtual Private Network system built with Linux and a Raspberry Pi 3.

- GRASSMARLIN
  Provides network situational awareness of Industrial Control Systems (ICS) and Supervisory Control and Data Acquisition (SCADA) networks to support network security.

(Those were in alphabetical order from 'A'... and I stopped at 'G')

HP Printer firmware updating
Also... many people have noted, tweeted, and sent photos showing that every HP Printer seen defaults to not updating firmware. HP must have made a decision that printers work out of the box and should not automatically update their firmware.

Errata

Ben (@Gingiraffe)
2 Things:
1) Apple 2FA doesn't use iMessage.
2) Apple 2FA does have some (limited) TOTP integration available.
Love the show, cheers!

Miscellany

I've figured out what it is that I love the Frontiers Saga:
I've started into book #8 of my re-read of the first 19.
There are many Sci-Fi genres of varying "believability" or "probability"... and don't get me wrong, just because something is far fetched it is not, in my opinion, any less enjoyable... except perhaps if it involves unicorns. (As I've mentioned before, I've never understood why wizards and spells and unicorns are s often lumped in with science fiction.) But, anyway... I love Peter Hamilton's work, Richard Philips, David Weber, Michael McCollum, Jack Campbell, Daniel Suarez. They are all entertaining. But something unique about the way Ryk Brown carries us along as he develops his story arc seems SO believable. His world doesn't have telepathy, beaming, cybernetic implants, co-resident AI, a gaia field linking all intelligence, or any of those common additives. It's just got real believable human people struggling to do the right thing in a setting of space.

SpinRite

Grant Taylor (@DrScriptt)
@SGgrc running SR before a format might speed up a format on an otherwise not-completely-happy drive.

I would say that's true, except that the explosion in storage capacity has changed the entire nature of formatting. It was once the case that "formatting a drive" meant first scanning and verifying the drive's storage surface looking for known and already marked defective sectors, or new problems. Then, the allocation clusters containing any defective sectors would be marked as "bad" and kept out of active use by the file system.

But, as SpinRite's own struggle to run in a practical length of time reveals, drives are becoming large relative to any system's ability to actually transfer all of the drive's data into the system. As we know, I'll be making a huge leap forward in SpinRite's performance with its next release. Through a combination of pure assembly language which will work directly with the hardware and maximum theoretical size data (32 megabyte buffers), SpinRite from v6.1 on will transfer data at the maximum rate limited only be the drive's rotation rate or, in the case of solid state media, the raw speed of the interface.

But today's drive sizes means that simple formatting is no longer able to check the drive's storage integrity. It must just assume everything is okay. The so-called "Quick Format" makes that assumption. It simply builds the file system's directory structure and hopes for the best. Since skipping any actual testing of the drive has simply been skipped for the sake of practicality and expediency, I would argue that running SpinRite SOMETIME -- whether before or after -- is

probably a good thing to do.

And note that this means, if you have NOT run SpinRite over your drive, there are almost certainly areas that have never, ever, been visited and checked and exercised.

Closing The Loop

Jeffry Erickson (@JeffryErickson)
@SGgrc ProtonMail now has a free (and paid) VPN. Topic for SN? ProtonMail @ProtonMail

#ProtonVPN is here. We have just launched a free VPN service to make secure Internet accessible to all. @ProtonVPN protonvpn.com

VPN services are more or less generic and widely available. That being the case, more choices and wider feature sets are better, but rapid adoption of a new service, just because it's new, is probably not the way to obtain the greatest security guarantee. A VPN service is one place where letting things settle down a bit make more sense -- all other things being equal. If a new service has some specific feature that makes it especially beneficial for an individual user, then that might outweigh a generic cautionary stance. But otherwise, it's likely wiser to go with something tried and true while a new service shakes out any startup glitches it might experience.

Dennis Thiel (@dennythetwit)
@SGgrc Hello Steve. What imaging software do you use for your daily images?
TeraByte's "Image for Windows" (and linux and DOS and with UEFI support.)

Simon Zerafa (@SimonZerafa)
@SGgrc Time to check for apps with access to important accounts.
Here's the list for Google. Do the same with Twitter and Facebook ??

Take a moment and review (and revoke) apps you've forgotten you gave Google access to:
https://myaccount.google.com/permissions

Our "say it isn't true" horrifying observation of the week:
Klingonveckan (@Tjoffex)
@SGgrc Regarding SN617 about 8 chars being the most common password length, I just have to bring up the fact that "password" is 8 chars...