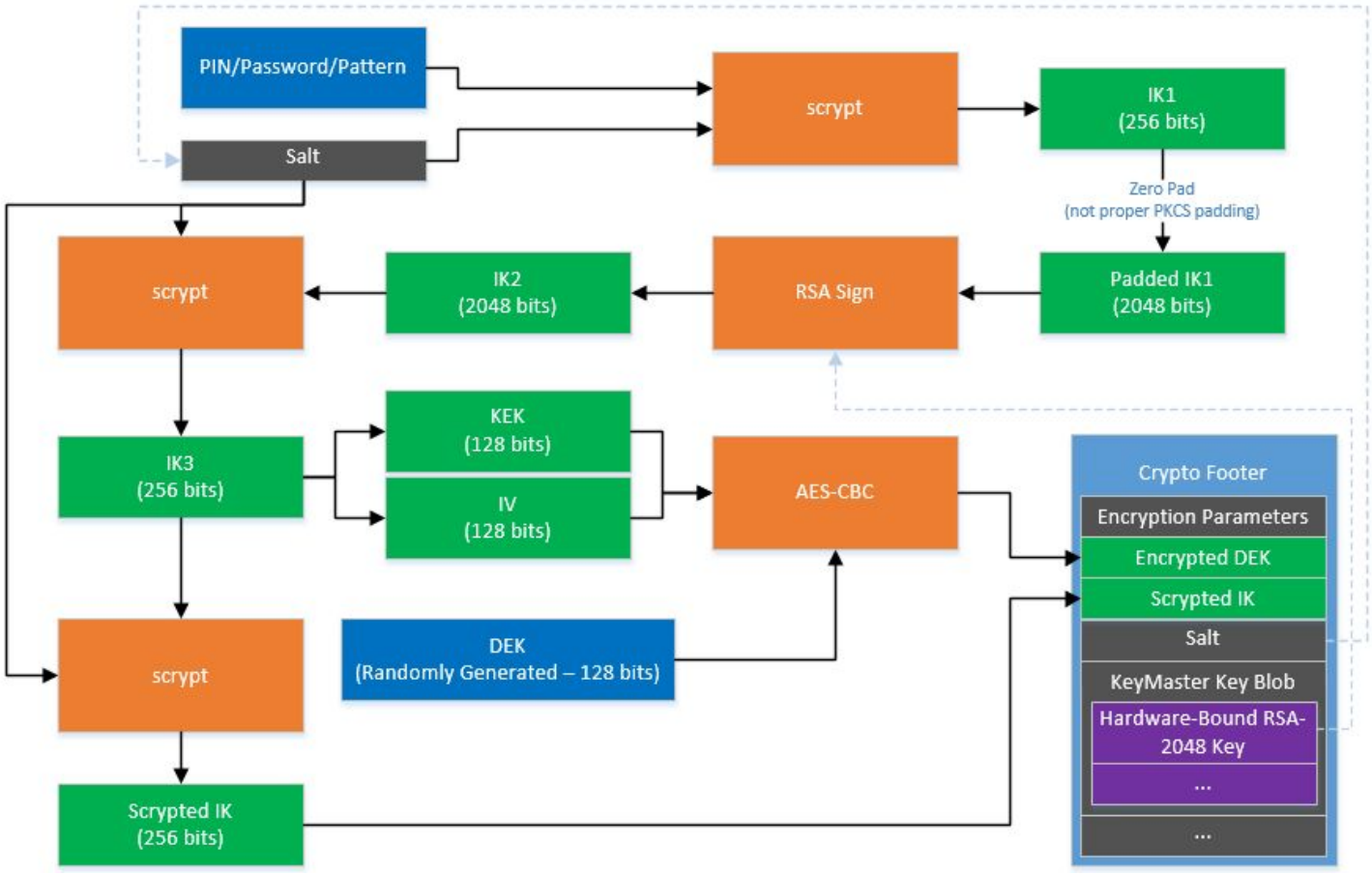
Security Now! #567 - 07-05-16

Hacking Certificates

This week on Security Now!

- Ransomware heading to a smartphone near you!
- A security researcher successfully extracts Android's FDE keys.
- Tavis @ Google's Project Zero finds horrific problems in Symantec/Norton products.
- That Brazilian judge is definitely NOT collecting Facebook "likes".
- What if you forget to re-register your router's setup domain name?
- This week's IoT Horror Story.
- A bit of errata and miscellany... then
- StartSSL/StartCom badly fumbles their Let's Encrypt clone, and a detailed look at certificate hacking.

Android's FDE Key Flow



Security News

Ransomware on mobile devices: knock-knock & block
https://usblog.kaspersky.com/mobile-ransomware-2016/7346/

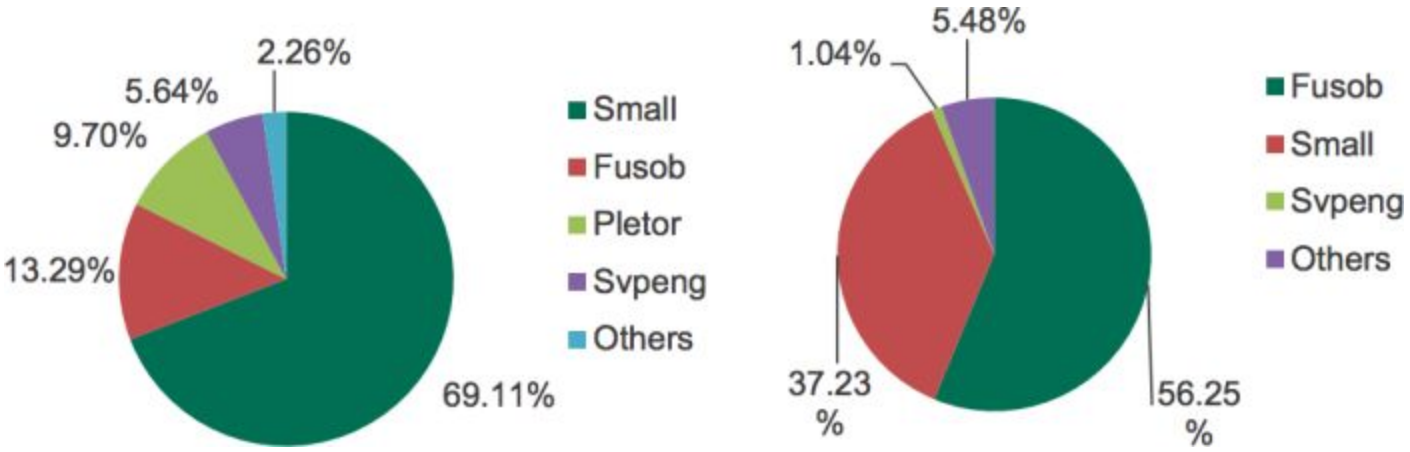
- Android customers infected with mobile ransomware hit 136K in April 2016, nearly quadrupling from March 2015

- Encrypt files or Block browser or the OS from working?

- Desktops favor file encryption, whereas mobile devices, which are typically backed up to the cloud, and could thereby have their files recovered -- and may also not contain any irreplaceable files -- are favoring the "Blocker"-style ransomware.

- Kaspersky writes: Blockers are the much more popular means to infect Android devices. On mobiles, they act simply by overlaying the interface of every app with their own, so a victim can't use any application at all. PC owners can get rid of a blocker with relative ease — all they need to do is remove the hard drive, plug it into another computer, and wipe out the blocker's files. But you can't simply remove the main storage from your phone — it's soldered onto the motherboard. That explains why blockers hold 99% of the mobile ransomware "market."

- In 2014–2015 four main actors dominated the mobile ransomware scene: Svpeng, Pletor, Small, and Fusob. At this time, Pletor has almost stopped its expansion; its creators have launched the infamous Acecard Trojan and seem to be pouring their resources into developing and spreading that instead. The developers of Svpeng have also refocused, mostly on the banking version of the Trojan. That leaves only two big mobile ransomware families, Small and Fusob. As a result, in 2015–2016 these two Trojans represent more than 93% of the mobile ransomware space.



- It's noteworthy that the Fusob and Small Trojan families have a lot in common. Both display fake screens that pretend to be signed by authorities and accuse the victims of misdemeanor. Both say that a criminal case will be opened unless the user pays the fine.

- Both Fusob and Small also offer rather strange ways to pay the ransom: Fusob suggests payment with iTunes gift cards, whereas Small offers victims the option of paying by the Kiwi payment system or with MoneyPak xpress Packet vouchers. Both were probably created by some Russian-speaking groups of cybercriminals, but with very different approaches.

- Fusob:
  Fusob first detects the device's language, and if it is one of the post-Soviet republic languages, Fusob simply does nothing. Otherwise, it shows a screen that claims to come from the NSA and demands ransom — from $100 to $200. The majority of Fusob's victims (more than 41%) live in Germany; the United Kingdom and the United States hold the second and third places, with 14.5% and 11.4%.

- Small:
  Then, there's the Small family. Almost 99% of Small's victims are located in three of the countries that Fusob avoids: Russia, Kazakhstan, and Ukraine. Small ransomware shows a government-themed screen with payment instructions, threats, and a demand for 700–3,500 rubles ($10 to $50) to unlock the infected device. An English-locale version of Small also exists — it has a different block-screen that mentions the FBI and demands about $300.

- There are two more versions of Small. One is a cryptolocker that performs the same operations the first version does, and on top of that encrypts the files on the device's SD card. The second is a multifunction Trojan that is capable of stealing money, siphoning data, and, of course, locking the device.

- What to do????

    - Install applications only from official shops such as Google Play. To be sure that no application makes its way onto your device from an untrusted source, go to Android settings, choose Security, and make sure that the "Unknown Sources" box is not checked.

    - Regularly update your device's firmware and its installed apps. You can choose to update apps automatically, but you still have to update the system manually — and it's better to do that as soon as an over-the-air (OTA) update arrives.

    - Install a strong security solution. (Kaspersky)
        - (I'm out of the mainstream on this one.)

Blog post title: Bits, Please! (Android FDE Key Extraction)
https://bits-please.blogspot.com/2016/06/extracting-qualcomms-keymaster-keys.html

- Extracting Qualcomm's KeyMaster Keys - Breaking Android Full Disk Encryption

- First let's recall how Apple's FDE - full disk encryption - works:
  - Hardware-based, factory-set, 256-bit UID (unique ID) key, which cannot be read or written... it can only be invoked without being exposed for AES crypto operations.
  - It is "Tangled" with the user's password to create a device-dependent key.
  - For data encrypted on that device, ONLY that device can be used for decryption.
    - (This is CRUCIAL for preventing removal of the keys for massive, hardware accelerated, parallel brute-force cracking of the user's password.
  - Hardware enforces an 80-millisecond non-short-circuitable "Tanglement" overhead.

- Now... the Android implementation:

  - Starting at v5.0, Android devices default to protecting all of the user's information by enabling full disk encryption.

  - Android's FDE is based upon a Linux Kernel subsystem known as "dm-crypt"

  - So dm-crypt is not "home grown", it's been widely deployed and researched.

  - HOWEVER, as we know, an encryption system implementation may have no weaknesses... but the vulnerability will be its key management. THAT's the hard part.

  - Also, research has shown that the comparative clunky keyboard of mobile devices induce their users to choose easier-to-enter lower-entropy passwords.

- Android devices implement a "TrustZone" with a secondary secure processor.

  - "Trustletts" can be loaded into this processor's execution space.

  - The code in the TrustZone can be extracted and reverse-engineered.

  - Several exploits have been found which break the TrustZone's trust.

    - However a patch for one was made available in January and for another in May.

    - But today ~37% of Android devices which were audited in the enterprise remain vulnerable.

- In this case, this researcher figured out how to extract the keys in order to perform high-speed hardware-accelerated brute force password cracking.

- "Jens Steube" -- the author of HashCat -- has offered to accelerate the cracking.

- The moral:
  - Making things difficult is no longer good enough. They need to be impossible.

High-severity bugs in 25 Symantec/Norton products imperil millions
(This is a classic "attack surface expansion")
http://arstechnica.com/security/2016/06/25-symantec-products-open-to-wormable-attack-by-unopened-e-mail-or-links/

- Dan Goodin's reporting in ArsTechnica begins: Much of the product line from security firm Symantec contains a raft of vulnerabilities that expose millions of consumers, small businesses, and large organizations to self-replicating attacks that take complete control of their computers, a researcher warned Tuesday.

- That researcher: Tavis Ormandy / Google's Project Zero
  - https://googleprojectzero.blogspot.com/2016/06/how-to-compromise-enterprise-endpoint.html

- Tavis' detailed blog post begins: Symantec is a popular vendor in the enterprise security market, their flagship product is Symantec Endpoint Protection. They sell various products using the same core engine in several markets, including a consumer version under the Norton brand.
  Today, (writes Tavis), we're publishing details of multiple critical vulnerabilities that we discovered, including many wormable remote code execution flaws.
  These vulnerabilities are as bad as it gets. They don't require any user interaction, they affect the default configuration, and the software runs at the highest privilege levels possible. In certain cases on Windows, vulnerable code is even loaded into the kernel, resulting in remote kernel memory corruption.
  As Symantec use the same core engine across their entire product line, all Symantec and Norton branded antivirus products are affected by these vulnerabilities, including:
  - Norton Security, Norton 360, and other legacy Norton products (All Platforms)
  - Symantec Endpoint Protection (All Versions, All Platforms)
  - Symantec Email Security (All Platforms)
  - Symantec Protection Engine (All Platforms)
  - Symantec Protection for SharePoint Servers
  - And all the rest, too.

- Some of these products cannot be automatically updated, and administrators must take immediate action to protect their networks. Symantec has published advisories for customers.

- In a coordinated disclosure, shortly before Tavis' post, Symantec issued its own advisory, which listed the 17 Symantec enterprise products and eight Norton consumer and small business products affected.

- https://www.symantec.com/security_response/securityupdates/detail.jsp?fid=security_advisory&pvid=security_advisory&year=&suid=20160628_00

- Tavis wrote: "Because Symantec uses a filter driver to intercept all system I/O, just emailing a file to a victim, or sending them a link to an exploit, is enough to trigger it - the victim does not need to open the file or interact with it in anyway. Because no interaction is necessary to exploit it, this is a wormable vulnerability with potentially devastating consequences to Norton and Symantec customers.

An attacker could easily compromise an entire enterprise fleet using a vulnerability like this. Network administrators should keep scenarios like this in mind when deciding to deploy Antivirus, it's a significant tradeoff in terms of increasing attack surface.

- The flaws reside in the engine the products use to reverse the compression malware developers use to conceal their malicious payloads. The unpackers work by parsing code contained in files before they're allowed to be downloaded or executed. Because Symantec runs the unpackers directly in the operating system kernel, errors can allow attackers to gain complete control over the vulnerable machine.

- Tavis wrote that one of the proof-of-concept exploits he devised works by exposing the unpacker to odd-sized records which cause inputs to be incorrectly rounded-up, resulting in a buffer overflow.

- A separate "decomposer library" included in the vulnerable software contained open-source code that in some cases hadn't been updated in at least seven years... despite the fact that vulnerabilities had been found in some of the aging code and the seven year old vulnerability disclosures were accompanied by publicly available exploits.

- ***The 10,000 foot view***: Microsoft has spent painful decades carefully hardening the core Windows operating system. Then Symantec comes along and installs a bug-ridden traffic-intercepting filter in the OS kernel which completely compromises Microsoft's hard-won security.

Brazilian judge freezes $6.07M in Facebook bank account after WhatsApp fails to turn over messages in drug case.
http://www.reuters.com/article/us-brazil-facebook-whatsapp-idUSKCN0ZH3EX

- Last Thursday, a court in Brazil, blocked >$6 million USD in Facebok , Inc. funds after the company's WhatsApp messaging service failed to turn over messages sought in a drug-related case.

- (This, of course, is PURE GOLD advertising for WhatsApp/Facebook, worth far more than $6mil.)

- After repeated failure over five months to turn over the information, Brazil's G1 news service reported that a judge froze the funds, which are equal to WhatsApp's accumulated fines for non-compliance in the case.

- Because WhatsApp has no bank accounts in Brazil, the judge froze funds owned by its parent, Facebook. The Brazilian court did not, though, again use provisions of Brazil's Internet law that allows courts to shut down service in some cases of non-compliance with court orders. Earlier this year a judge ordered a 72-hour shutdown of WhatsApp, which angering many of the service's 100 million Brazilian users. That shutdown was lifted after about 24 hours by another court.

What if you forget to re-register your router's setup domain name?
https://www.helpnetsecurity.com/2016/07/05/tp-link-config-domains/

- TP-Link -- extremely popular consumer and SOHO router vendor
  - Was once using "tplinklogin.net"
  - Some time ago switched to using "tplinkwifi.net"
  - But the LABELS on, and manuals for, all of their older routers still refer to "tplinklogin.net."
  - THAT domain registration expired, it was acquired... and now has a price of $2.5 million.
  - The bad news is that it is a vector for malware distribution into routers!

- BUT... our friend Michael Horowitz, who writes the Defensive Computing column for InfoWorld, researched this further and verified that for any users BEHIND the router, even if the router's LAN address has been reconfigured, any INTERNAL reference to "tplinklogin.net" will be intercepted BY THE ROUTER and the user will simply be taken to the router's local login page for configuration.

- User's OUTSIDE of the router's LAN *could* be fooled, but the problem is not quite a big as it might otherwise have been.

- NOTE: Michael operates "RouterSecurity.org" and strongly recommends that savvy users simply avoid all of the consumer "blue box" routers. He recommends the Pepwave Surf SOHO from Peplink. Michael writes: "I maintain a long writeup of its pros/cons at RouterSecurity.org. The Surf SOHO is a business class router that is a big step up from consumer models, yet is reasonably priced and no harder to configure than the average consumer targeted router. My only relationship with Peplink is that of a customer."
  - http://www.peplink.com/products/pepwave-surf-soho/
  - http://routersecurity.org/pepwavesurfsofo.php

This week's IoT horror story:
https://techcrunch.com/2016/07/01/security-researcher-gets-threats-over-amazon-review/
Security researcher Matthew Garrett writes a (one-star) Amazon review for a IoT AC plug titled:

" Nice hardware, infuriating setup issues, terrible insecure software "...

There's a lot to like about this hardware, but unfortunately it's entirely overwhelmed by everything there is to hate about it. But before we get to that: I received this product at a discount in return for writing an honest review of it. Onwards!

The packaging is entirely reasonable. There's a small cardboard box, a sheet of instructions, and a piece of hardware securely wrapped in bubblewrap. It's very small, but feels well built. There's no creaking plastic under pressure, and no parts feel loose. Once it's plugged in a blue LED in the button starts blinking, waiting for you to set it up. The app attempts to walk you through the setup, but things start going wrong here.

This system is based on the ESP8266 module, which is a great choice for this sort of application. It's cheap but well featured, and there's a lot of easy off the shelf code that vendors can

incorporate into it to cut down development time and ship better products faster. One of the features available is something called "Smart connect", where an app on a phone encodes your wifi password into network broadcasts of different lengths. It's possible to detect these even without the password, so this allows your phone to pass the information on to the socket without having to fiddle about connecting to a different network. Simply hold down the power button to enter setup mode, and the phone does the rest.

At least, it does in theory. In practice it fails for two reasons - the first is that it'll happily try to do this on a 5GHz network, even though the socket only has 2.4GHz support. The other is that the app doesn't have the appropriate permissions to do this on Android 6, so it doesn't work on new Android phones even if you are on a 2.4GHz network. However, it also supports a more traditional setup mode. By holding down the power button again, it turns into an access point. You connect to it and the app sends the network data. Simple.

Again, at least, in theory. In practice the app is looking for a network called "SmartPlug" and this version of the hardware creates a network called "XW-G03", so it never finds it. I ended up reverse engineering the app in order to find out the configuration packet format, sent it myself and finally had the socket on the network. This is, needless to say, not a reasonable thing to expect average users to do. The alternative is to find an older Android device or use an iPhone to do the setup.

Once it's working, you can just hit a button on the app and your socket turns on or off. You can also program a timer. If your phone is connected to the same network as the socket then this is just done by sending a command directly, but if not you send a command via an intermediate server in China (the socket connects to the server when it joins the wireless and then waits for commands). The command packets look like they're encrypted, but in reality there's no real cryptography here at all. I wrote a simple program to decode the packets and looked at them in more detail. There's a lot of unnecessary complexity in the packet format, but in the end the relevant things are just a command and the MAC address of the socket. On the local network this is sent directly to the socket, otherwise it goes via the server in China. The server looks at the address and uses that to decide which socket to pass it on to. Other than the destination, the packets are identical.

This is a huge problem. If anybody knows the MAC address of one of your sockets, they can control it from anywhere in the world. You can't set a password to stop them, and a normal home router configuration won't block this. You need to explicitly firewall off the server (it's 115.28.45.50) in order to protect yourself. Again, this is completely unrealistic to expect for a home user, and if you do this then you'll also entirely lose the ability to control the device from outside your home.

In summary: by default this is stupendously insecure, there's no reasonable way to make it secure, and if you do make it secure then it's much less useful than it's supposed to be. Don't buy it.

(continuing)...

TechCrunch's Kate Conger reached out to Matthew.
She writes: “Garrett sent me a few of the emails he received from the company.”

“Just now my boss has blamed me, and he said if I do not remove this bad review, he will quit me. Please help me,” the representative wrote. “Could you please change your bad review into good?”

Kate writes: Garrett responded that he would update the review if the manufacturer fixed the flaw. The AuYou representative insisted she would be fired if the review was not updated. A week later, she followed up again, asking Garrett to take down the review. The representative then said that she would report Garrett to Amazon if he didn’t take down the review, and that other Amazon reviewers had written in to complain about it.

Garrett says he leaves a lot of security based reviews on Amazon and this has never happened to him. Of course, no one needs to lose their job over a single Amazon review. Garrett says he’s not sure if he’s being manipulated or if someone’s job is really on the line.

“If I thought that there was a realistic chance that people were going to lose their jobs over something I was writing, that’s something that would make me reconsider,” he says. “On the other hand, the attitude that many companies have of not giving any indication of caring about the security of the people they’re selling to is horrifying in its own way. That is important — to make people aware when choosing these devices.”

- The plug appears to have been taken down from Amazon.
- Matthew's review remains with a permalink. But the plug could not be found.
- (AuYou is also offering two full home security systems <<shudder>>)

Errata

- Steve doesn't use version control.
- "Penultimate" SoftICE debugger for DOS.

Miscellany

Killjoys & Dark Matter have both returned to SyFy (post SyFy reboot.)

The Atlantic, July/August:

- How American Politics Went Insane, by Jonathan Rauch.
- "It happened gradually—and until the U.S. figures out how to treat the problem, it will only get worse."
- http://bit.ly/sn-567
- https://media.grc.com/How_American_Politics_Went_Insane.pdf

First edition of the 2nd-generation HSF page is online.

- (iHerb's stock was almost immediately drained.)

SpinRite

Jeremy Leik (pronounced "like" - blame my German ancestors) in Dimondale, Michigan
Subject: SpinRite mentioned on Technibble.com
Date: 24 Jun 2016 08:14:39
:
Hello Steve (and Leo)

I'm writing today because I'm a user of SpinRite, and a believer in its capabilities. It has saved a few drives for me, and allowed me to recover data from a RAID array that came out of a NAS with failed electronics.

I came across an article called "Technical Overview of Popular Software Data Recovery Procedures" over at Technibble. They mention SpinRite on p2 of their article. The part where they describe SpinRite seems pretty simplistic, and lumps it in with other software tools. I would be interested in hearing from the horses mouth (so to speak) what your thoughts are on this subject.

Thanks so much.
Jeremy Leik (pronounced "like" - blame my German ancestors).

StartCom tries to follow Let's Encrypt... and fails badly!

StartCom, the CA (Certificate Authority) behind the StartSSL service, launched the StartEncrypt project on June 4, inspired by the success of the Let's Encrypt project.
https://www.computest.nl/blog/startencrypt-considered-harmful-today/

CompuTest's disclosure of this is almost comically understated:

Recently, one of our hackers found a critical vulnerability in StartCom's new StartEncrypt tool, that allows an attacker to gain valid SSL certificates for domains he does not control. While there are some restrictions on what domains the attack can be applied to, domains where the attack will work include google.com, facebook.com, live.com, dropbox.com and others.

StartCom, known for its CA service under the name of StartSSL, has recently released the StartEncrypt tool. Modeled after LetsEncrypt, this service allows for the easy and free installation of SSL certificates on servers. In the current age of surveillance and cybercrime, this is a great step forwards, since it enables website owners to provide their visitors with better security at small effort and no cost.

However, there is a lot that can go wrong with the automated issuance of SSL certificates. Before someone is issued a certificate for their domain, say computest.nl, the CA needs to check that the request is done by someone who is actually in control of the domain. For "Extended Validation" certificates this involves a lot of paperwork and manual checking, but for simple, so-called "Domain Validated" certificates, often an automated check is used by sending an email to the domain or asking the user to upload a file. The CA has a lot of freedom in how the check

is performed, but ultimately, the requester is provided with a certificate that provides the same security no matter which CA issued it.

In order to make the issuance of certificates easy, this tool runs on your server (Windows or Linux), detects your webserver configuration, and requests DV certificates for the domains that were found in your config. Then, the StartCom API does a HTTP request to the website at the domain you requested a certificate for, and checks for the presence of a piece of proof that you have access to that website. If the proof is found, the API returns a certificate to the client, which then installs it in your config.

However, it appears that the StartEncrypt tool did not receive proper attention from security-minded people in the design and implementation phases. While the client contains numerous vulnerabilities, one in particular allows the attacker to "trick" the validation step.

The first bug

The API checks if a user is authorized to obtain a certificate for a domain by downloading a signature from that domain, by default from the path "/signfile". However, the client chooses that path during a HTTP POST request to that API. A malicious client can specify a path to any file on the server for which a certificate is requested. This means that, for example, anyone can obtain a certificate for sites like dropbox.com and github.com where users can upload their own files.

That's not all

While this is serious, most websites don't allow you to upload a file and then have it presented back to you in raw format like github and dropbox do. Another issue in the API however allows for much wider exploitation of this issue: the API follows redirects. So, if the URL specified in the "verifyRes" parameter returns a redirect to another URL, the API will follow that until it gets to the proof. Even if the redirect goes off-domain. Since the first redirect was to the domain that is being verified, the API considers the proof correct even if it is redirected to a different website.

This means that an attacker can obtain an SSL certificate for any website that either:

- Allows users to upload files and serves them back raw, or
- Has an "open redirect" vulnerafeature in it

Open redirects are pages that take a parameter that contains a URL, and redirect the user to it. This seems like a strange feature, but this mechanism is often implemented for instance in logout or login pages. After a successful logout, you will be redirected to some other page. That other page URL is included as a parameter to the logout page.

For instance, http://mywebsite/logout?returnURL=/login might redirect you to /login after logout.

While many see open redirects as a vulnerability that needs fixing, others do not. Google for instance has excluded open redirects from their vulnerability reward program. By itself an open redirect is not exploitable, so it is understandable that many do not see it as a security issue.

However, when combined with a vulnerability like the one present in StartEncrypt, the open redirect suddenly has great impact.

It's actually even worse: the OAuth 2.0 specification practically mandates that an open redirect must be present in each implementation of the spec. For this reason, login.live.com and graph.facebook.com for instance contain open redirects.

When combining the path-bug with the open redirect, suddenly many more certificates can be obtained, like for google.com, paypal.com, linkedin.com, login.live.com and all those other websites with open redirects. While not every website has an open redirect feature, many do at some point in time.

That's still not all

Apart from the vulnerability described above, we found some other vulnerabilities in the client while doing just a cursory check. We are only publishing those that according to StartCom have been fixed or are no issue. These include:

- The client doesn't check the server's certificate for validity when connecting to the API, which is pretty ironic for an SSL tool. [Steve: ...allowing for man-in-the-middle attacks against the StartEncrypt client. Bad guys could arrange to have invalid certificates issued... thus creating a denial-of-service against targeted sites using StartEncrypt.]

- The API doesn't verify the content-type of the file it downloads for verification, so attackers can obtain certificates for any website where users can upload their own avatars (in combination with the above vulnerabilities of course)

- The private key on the server (/usr/local/StartEncrypt/conf/cert/tokenpri.key) is saved with mode 0666, so world-readable, which means any local process or user can read or modify it.

All in all, it doesn't seem like a lot of attention was paid to security in the design and implementation of this piece of software.

Time permitting…

- **Revisiting the power of Certificate pinning.**
- The special position Google has being their own CA (Intermediate CA signed by GeoTrust) and having the most-used web browser on the Internet...