← Back to Disclosures

19 Zero-Day Vulnerabilities Amplified by the Supply Chain

Overview- Ripple20

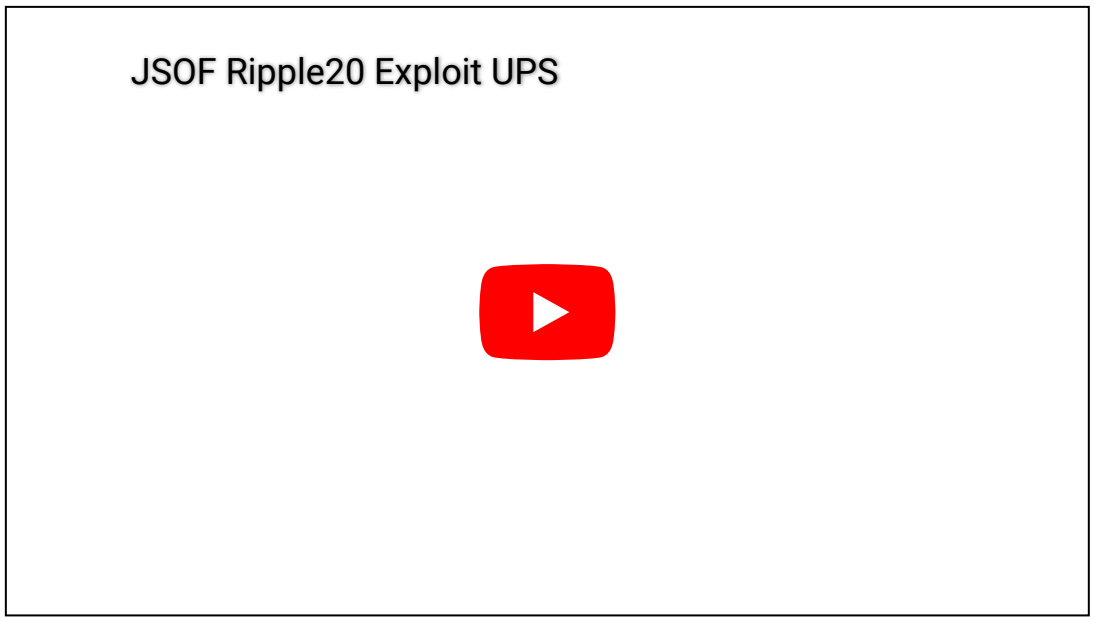
The JSOF research lab has discovered a series of zero-day vulnerabilities in a widely used low-level TCP/IP software library developed by Treck, Inc. The 19 vulnerabilities, given the name ***Ripple20***, affect hundreds of millions of devices (or more) and include multiple remote code execution vulnerabilities. The risks inherent in this situation are high. Just a few examples: data could be stolen off of a printer, an infusion pump behavior changed, or industrial control devices could be made to malfunction. An attacker could hide malicious code within embedded devices for years. One of the vulnerabilities could enable entry from outside into the network boundaries; and this is only a small taste of the potential risks.

The interesting thing about ***Ripple20*** is the incredible extent of its impact, magnified by the supply chain factor. The wide-spread dissemination of the software library (and its internal vulnerabilities) was a natural consequence of the supply chain “ripple-effect”. A single vulnerable component, though it may be relatively small in and of itself, can ripple outward to impact a wide range of industries, applications, companies, and people.

Ripple20 reached critical IoT devices from a wide range of fields, involving a diverse group of vendors. Affected vendors range from one-person boutique shops to Fortune 500 multinational corporations, including HP, Schneider Electric, Intel, Rockwell Automation, Caterpillar, Baxter, as well as many other major international vendors

A detailed technical report of two of the vulnerabilities and their exploitation can be found in the **CVE-2020-11896/CVE-2020-11898 whitepaper** (click here).

JSOF has demonstrated exploitation of these vulnerabilities on different devices as a proof-of-concept. Watch us turn off the plug on a UPS:



JSOF will be providing scripts for the identification of products running Treck upon request.
For more information or requests please contact: Ripple20@jsof-tech.com

Following the software trail

The software library spread far and wide, to the point that tracking it down has been a major challenge. As we traced through the distribution trail of Treck's TCP/IP library , we discovered that over the past two decades this basic piece of networking software has been spreading around the world, through both direct and indirect use. As a dissemination vector, the complex supply chain provides the perfect channel, making it possible for the original vulnerability to infiltrate and camouflage itself almost endlessly.

We even discovered different branches in different geographic areas. Back in the 1990s, Treck collaborated with a Japanese company named Elmic Systems. They later split apart and went their separate ways. This resulted in two separate branches of the TCP/IP stack devices – one managed by Treck and one managed by Elmic Systems – marketed in totally separate areas, with no contact between them.

Other than ELMIC, the Treck stack could also be known by other names: Net+ OS, Quadnet, GHNET v2, Kwiknet.

JSOF has been invited to speak about these vulnerabilities at Black Hat USA, August 2020.

More Information

For a detailed description of the supply chain effect and impact of ***Ripple20*** and how JSOF tracked it down, see Supply chain and Disclosure.

For a technical overview, including a full list of affected vendors take a look at the Technical Overview section.
Advisories by: ICS CERT, CERT/CC, JPCERT/CC,CERT-IL
Advisories by: ABB, Aruba Networks, B.Braun, Baxter, CareStream, Caterpillar, Cisco ,Digi International, Green Hills, HP, HPE, Intel, Miele, Mitsubishi Electric, Opto22, Rockwell Automation, Schneider Electric, Smiths Medical, Teradici, Xerox

For Mitigations and Risk Evaluation, see Risk & Mitigation section.

Thank you

This was a big project. It would not be possible without the help of many.

• **Andrew** and **Mark** from EFF for their time and patience
• **Elad Luz** from CyberMDX for support, advice, comments, finding affected devices and being great overall.
• **Claroty** for their much-needed help and support
• **CISA ICS-CERT, CER/TCC, JPCERT/CC CERT-IL** for all their help and support coordinating the disclosure
• **Zack Weiner** for Public Relations and communications
• **Gavin Goodvach** for video editing, and professional censorship
• **Daniel Dos Santos** from Forescout for finding affected devices
• **Nate Pollack** and **Jon Rabinowitz** for communications help
• **Shaked Heyman** for holding tight while his mom was away making the world more secure

• **lyrebirds.dk** for sharing some firmwares (not vulnerable)
• **Shani** and **Hadas** for building our new website under a crazy timeline

Credits

• **Research:** Moshe Kol, Ariel Schon, Shlomi Oberman, Alon Dotan , Andrey Zagrebin, Yuli Shapiro
• **Orchestration & oversight:** Sari Heyman, Shlomi Oberman
• **Keeping the company afloat while Shlomi is occupied:** Yuli Shapiro & Team
• **Giving us grief:** vendors worldwide
• **Using obscure variations of x86 in your products:** you know who you are.

Risk Evaluation and Mitigations

Ripple20 poses a significant risk from the devices still in use. Potential risk scenarios include:

• An attacker from outside the network taking control over a device within the network, if internet facing.

• An attacker who has already managed to infiltrate a network can use the library vulnerabilities to target specific devices within it.

• An attacker could broadcast an attack capable of taking over all impacted devices in the network simultaneously.

• An attacker may utilize affected device as a way to remain hidden within the network for years

• A sophisticated attacker can potentially perform an attack on a device within the network, from outside the network boundaries, thus bypassing NAT configurations. This can be done by performing a MITM attack or a dns cache poisoning.

• In some scenarios, an attacker may be able to perform attacks from outside the network by replying to packets that leave network boundaries, bypassing NAT

In all scenarios, an attacker can gain complete control over the targeted device remotely, with no user interaction required.

JSOF recommends taking measures to minimize or mitigate the risk of device exploitation. Mitigation options depend on the context. Device vendors would have different approaches from network operators. In general, we recommend the following steps:

- All organizations must perform a comprehensive risk assessment before deploying defensive measures.
- First deploy defensive measures in a passive “alert” mode.
- Mitigation for device vendors:
  - Determine if you use a vulnerable Treck stack
  - Contact Treck to understand risks
  - Update to latest Treck stack version (6.0.1.67 or higher)

(based on CERT/CC and CISA ICS-CERT advisories)

- The first and best mitigation is updating to patched versions of all devices.
- If devices cannot be updated, the following steps are recommended:
  - Minimize network exposure for embedded and critical devices, keeping exposure to the minimum necessary, and ensuring that devices are not accessible from the Internet unless absolutely essential.
  - Segregate OT networks and devices behind firewalls and isolate them from the business network.
  - Enable only secure remote access methods.
- Block anomalous IP traffic.
- Block network attacks via deep packet inspection, to reduce risk to your Treck embedded TCP/IP-enabled devices.

Pre-emptive traffic filtering is an effective technique that can be applied as appropriate to your network environment. Filtering options include:

• Normalize or block IP fragments, if not supported in your environment.
• Disable or block IP tunneling (IPv6-in-IPv4 or IP-in-IP tunneling), if not required.
• Block IP source routing, and any IPv6 deprecated features, like routing headers VU#267289
• Enforced TCP inspection, rejecting malformed TCP packets.
• Block unused ICMP control messages, such as MTU update and Address Mask updates.
• Normalize DNS through a secure recursive server or DNS inspection firewall. (Verify that your recursive DNS server normalizes requests.)
• Provide DHCP/DHCPv6 security, with features such as DHCP snooping.
• Disable/Block IPv6 multicast capabilities if not used in the switching infrastructure.
• Disable DHCP where static IPs can be used.
• Employ network IDS and IPS signatures.

• Employ network segmentation, if available.

Technical overview

Ripple20 is a set of 19 vulnerabilities found on the Treck TCP/IP stack . Four of the Ripple20 vulnerabilities are rated critical, with CVSS scores over 9 and enable Remote Code Execution. One of the critical vulnerabilities is in the DNS protocol and may potentially be exploitable by a sophisticated attacker over the internet, from outside the network boundaries, even on devices that are not connected to the internet.

A second Whitepaper, to be released following BlackHat USA 2020 will be detailing the exploitation of CVE-2020-11901, a DNS vulnerability, on a Schneider Electric APC UPS device. The other 15 vulnerabilities are in ranging degrees of severity with CVSS score ranging from 3.1 to 8.2, and effects ranging from Denial of Service to potential Remote Code Execution.

Most of the vulnerabilities are true Zero-days, with 4 of them having been closed over the years as part of routine code changes, but remained open in some of the affected devices (3 lower severity, 1 higher). Many of the vulnerabilities have several variants due to the Stack configurability and code changes over the years.

stack implementations and usually had to do with RFC misinterpretations or deprecated RFCs.

Ripple20 vulnerabilities are unique both in their widespread effect and impact due to supply chain effect and being vulnerabilities allowing attackers to bypass NAT and firewalls and take control of devices undetected, with no user interaction required. This is due to the vulnerabilities being in a low level TCP/IP stack, and the fact that for many of the vulnerabilities, the packets sent are very similar to valid packets, or, in some cases are completely valid packets. This enables the attack to pass as legitimate traffic.

A white paper describing a third vulnerability and exploitation on the UPS will be released following Black Hat USA 2020.

Vulnerabilities include:

CVE ID	CVSSv3	Description	Potential Impact	Fixed on Version
CVE-2020-11896	10	This vulnerability can be triggered by sending multiple malformed IPv4 packets to a device supporting IPv4 tunneling. It affects any device running Treck with a specific configuration. It can allow a stable remote code execution and has been demonstrated on a Digi International device. Variants of this Issue can be triggered to cause a Denial of Service or a persistent Denial of Service, requiring a hard reset.	Remote Code Execution	6.0.1.66 (release 30/03/2020)
CVE-2020-11897	10	This vulnerability can be triggered by sending multiple malformed IPv6 packets to a device. It affects any device running an older version of Treck with IPv6 support, and was previously fixed as a routine code change. It can potentially allow a stable remote code execution.	Out-of-Bounds Write	5.0.1.35 (release 04/06/2009)
CVE-2020-11901	9	This vulnerability can be triggered by answering a single DNS request made from the device. It affects any device running Treck with DNS support and we have demonstrated that it can be used to perform Remote Code Execution on a Schneider Electric APC UPS. In our opinion this is the most severe of the vulnerabilities despite having a CVSS score of 9.0, due to the fact that DNS requests may leave the network in which the device is located, and a sophisticated attacker may be able to use this vulnerability to take over a device from outside the network through DNS cache poisoning, or other methods. Thus an attacker can infiltrate the network and take over the device with one vulnerability bypassing any security measures. The malformed packet is almost completely RFC compliant, and so it will likely be difficult for security products such as firewalls to detect this vulnerability. On very old versions of the Treck stack, still running on some devices, the transaction ID is not randomized making the attack easier.	Remote Code Execution	6.0.1.66 (release 03/03/2020)

Additional vulnerabilities are listed below.

CVE ID	CVSSv3 Score	Description	Fixed In Version
CVE-2020-11898	9.1	Improper Handling of Length Parameter Inconsistency (CWE-130) in IPv4/ICMPv4 component, when handling a packet sent by an unauthorized network attacker. Possible Exposure of Sensitive Information (CWE-200)	6.0.1.66 (release 03/03/2020)
CVE-2020-11900	8.2	Possible Double Free (CWE-415) in IPv4 tunneling component when handling a packet sent by a network attacker. Use After Free (CWE-416)	6.0.1.41 (release 10/15/2014)
CVE-2020-11902	7.3	Improper Input Validation (CWE-20) in IPv6OverIPv4 tunneling component when handling a packet sent by an unauthorized network attacker. Possible Out-of-bounds Read (CWE-125)	6.0.1.66 (release 03/03/20)
CVE-2020-11904	5.6	Possible Integer Overflow or Wraparound (CWE-190) in Memory Allocation component when handling a packet sent by an unauthorized network attacker Possible Out-of-Bounds Write (CWE-787)	6.0.1.66 (release 03/03/2020)
CVE-2020-11899	5.4	Improper Input Validation (CWE-20) in IPv6 component when handling a packet sent by an unauthorized network attacker. Possible Out-of-bounds Read (CWE-125), and Possible Denial of Service.	6.0.1.66 (release 03/03/20)
CVE-2020-11903	5.3	Possible Out-of-bounds Read (CWE-125) in DHCP component when handling a packet sent by an unauthorized network attacker. Possible Exposure of Sensitive Information (CWE-200)	6.0.1.28 (release 10/10/12)
CVE-2020-11905	5.3	Possible Out-of-bounds Read (CWE-125) in DHCPv6 component when handling a packet sent by an unauthorized network attacker. Possible Exposure of Sensitive Information (CWE-200)	6.0.1.66 (release 03/03/20)

CVE-2020-11907	5	Improper Handling of Length Parameter Inconsistency (CWE-130) in TCP component, from a packet sent by an unauthorized network attacker Integer Underflow (CWE-191)	6.0.1.66 (release 03/03/20)
CVE-2020-11909	3.7	Improper Input Validation (CWE-20) in IPv4 component when handling a packet sent by an unauthorized network attacker. Integer Underflow (CWE-191)	6.0.1.66 (release 03/03/20)
CVE-2020-11910	3.7	Improper Input Validation (CWE-20) in ICMPv4 component when handling a packet sent by an unauthorized network attacker. Possible Out-of-bounds Read (CWE-125)	6.0.1.66 (release 03/03/20)
CVE-2020-11911	3.7	Improper Access Control (CWE-284) in ICMPv4 component when handling a packet sent by an unauthorized network attacker. Incorrect Permission Assignment for Critical Resource (CWE-732)	6.0.1.66 (release 03/03/20)
CVE-2020-11912	3.7	Improper Input Validation (CWE-20) in TCP component when handling a packet sent by an unauthorized network attacker. Possible Out-of-bounds Read (CWE-125)	6.0.1.66 (release 03/03/20)
CVE-2020-11913	3.7	Improper Input Validation (CWE-20) in IPv6 component when handling a packet sent by an unauthorized network attacker. Possible Out-of-bounds Read (CWE-125)	6.0.1.66 (release 03/03/20)
CVE-2020-11914	3.1	"Improper Input Validation (CWE-20) in ARP component when handling a packet sent by an unauthorized network attacker." Possible Out-of-bounds Read (CWE-125)	6.0.1.66 (release 03/03/20)
CVE-2020-11908	3.1	Improper Null Termination (CWE-170) in DHCP component when handling a packet sent by an unauthorized network attacker. Possible Exposure of Sensitive Information (CWE-200)	4.7.1.27 (release 11/08/07)

Affected Vendors

The list of vendors has been assembled carefully by different means and represents vendors that may be affected. The list only contains vendors that CISA ICS-CERT has listed in an internal document as having been contacted.

The status of each vendor internal investigation is supplied by CISA ICS-CERT according to vendor response. Vendors stating they are not affected are also recorded for accountability purposes and future inquiries if they are raised again as potentially having affected products.

The list of affected products can be found in the vendor advisory linked above or directly from the vendor.
The list will be updated from time to time. Any vendor that would like to report a different status or believes there is a mistake can do so through a coordination agency or by emailing Ripple20@jsof-tech.com

The following vendors are affected or might be affected.
Latest update of vendors list: October 25 , 2020; 03:50 am ET

- Status: Confirmed (31)
- Status: Pending (66)
- Status: Not Affected (29)
- Status: Affected- Low Risk (4)

Supply Chain

To effectively address the dangers of ***Ripple20***, the widespread distribution pattern must be painstakingly traced through the complex supply chain. During the disclosure

appreciate just how extensive the supply chain picture really was. During the disclosure process we were also notified that there are 2 additional vulnerabilities disclosed anonymously at the same time. These are part of the 19 Ripple20 vulnerabilities, and all parties involved have been handling these vulnerabilities as part of the same disclosure process, and there are no further actions for vendors to take.

We could track the supply chain trails, but we needed to work with international organizations to extend our reach within organizations and domains for which we had no access. This is why the ***Ripple20*** disclosure process is being coordinated and overseen by multiple national computer emergency response team (CERT) organizations and regulators. We have also been collaborating with other security vendors. As can be understood from the scale of the problem, teamwork here is essential, and benefits all players.

Once ***Ripple20*** is publicized, we are hoping to increase awareness, enabling more companies to use the information to determine for themselves if their products, or products they are using, are vulnerable.

We will be releasing a blog post dedicated to the methods we used to track the supply chain and discover its complexities.

Vulnerability: dependent on the affected industry or sectors

In the case of ***Ripple20***, the starting point was embedded into Treck's TCP/IP low-level Internet protocol suite library. The library could be used as-is, configured for a wide range of uses, or incorporated into a larger library. The user could buy the library in source code format and edit it extensively. It can be incorporated into the code and implanted into a wide range of device types. The original purchaser could decide to rebrand, or could be acquired by a different corporation, with the original library history lost in company archives. Over time, the original library component could become virtually unrecognizable. This is why, long after the original vulnerability was identified and patched, vulnerabilities may still remain in the field, since tracing the supply chain trail may be practically impossible.

JSOF has conducted extensive, in-depth analysis, over many months, of the vendors affected by the Treck Internet protocol library vulnerability. The first challenge we experienced was simply being able to identify the relevant vendors Vendor identity could be obscured through the intricacies of the supply chain. Even when the vendors are identified, patch implementation is complex and not always possible. Over the course of the disclosure process we found that while patching was difficult for some vendors, it could potentially be even more difficult or close to impossible for some end users to install the patches. (For example, if the library is on a separate physical component or the company that produced the component has ceased operations.)

The number of devices that contain the vulnerable code base library is only a preliminary estimate; the number may realistically be in the billions.

Disclosure

From the beginning

some troubling data on potential vulnerabilities and contacted Treck to share our information as well as kick off a coordinated vulnerability disclosure (CVD) process. We understood from the beginning that Treck worked with many different vendors, with the potential to impact a large number of users. However, we simply had no idea of the scale and sheer magnitude of the situation, nor how complex the supply chain had become.

Our outreach to Treck was initially challenging. Treck is a small company serving a niche clientele, and not generally available for individual consumer approach. They also never appear to have been the target of independent security research. Eventually, we managed to track down an email contact at Treck and received an introduction. We notified them that we found a critical security vulnerability in their TCP/IP stack library, and it was our preference to work directly with them. Initially, we were faced with minimal communication. We later learned that Treck was assessing the information internally and with their legal advisors. (We also noticed litigation lawyers checking us out on LinkedIn and elsewhere).

Subsequently, we started to contact a few select vendors (such as Digi, HP, Intel, and Quadros) that we knew used Treck products and its vulnerable TCP/IP stack library, and asked them to help us make the appropriate connection with Treck. After this vendor outreach, the ball started rolling with Treck.

Working with Treck

Once Treck understood the gravity of the vulnerabilities, we worked cooperatively together, and JSOF provided them with descriptions of the vulnerabilities and some other information.

It was important to us that Treck would make sure their clients were notified of the vulnerabilities at hand. Due to NDAs and other complexities, Treck was not able to provide us with a list of its clients and users of the code library. This was our first brush with some of the supply chain challenges. As a result, Treck ended up taking the lead on the disclosure process, as they were best suited to notify their clients and provide the appropriate patches.

Interestingly, the process of remediating the vulnerability led some of Treck's clients to renew support contracts, thus Treck seems to have ended up profiting from the situation. When these companies were notified about ***Ripple20*** and well understood the potential risks, they soon realized the necessity of ongoing maintenance and importance of access to patches. In the end, many of the companies worked with Treck to either renew their contracts or make other arrangements. This is a thought-provoking lesson for both small and large vendors wary of facing security issues. A proactive security response approach means clients have a reason to pay for maintenance and support in order to keep the software up to date.

Standard industry practice is not to publicize a vulnerability until there is a patch available to fix it. We agreed on a standard 90-day period within which Treck would fix the vulnerabilities and notify their clients of the patch. Treck had the patch available around the end of March – 45 days before the 90-day deadline, and informed us that they would be reaching out to all affected clients to inform them of these vulnerabilities.

of consideration for these companies, the time period was extended from 90 days to over 120 days. Even so, some of the participating companies became difficult to deal with, as they made extra demands and some, from our perspective, seemed much more concerned with their brand's image than with patching the vulnerabilities.

After some deliberation, June 16 was the date chosen for publicizing ***Ripple20***. In the interim, once we understood the extent of the vulnerabilities and the sheer numbers involved, we focused on how to best use our time (120 days of patience) to identify and help to address all of the parties that could be at risk.

An international effort

Since Treck was unable to supply us with a comprehensive list of their clientele, we decided to create our own, in order to find out who was affected and ensure that everyone was notified and would have the vulnerability patched. We used a creative supply chain tracking approach, in an attempt to understand the scale of distribution and the extent of the problem. We soon began to realize and appreciate just how complex the supply chain picture really was.

It became clear to us that tracking the extent of Treck library distribution was too large for just one small team. We could track the supply chain trails, but we needed to work with international organizations to extend our reach within organizations and domains for which we had no access.

This is why the ***Ripple20*** disclosure process is being coordinated and overseen by multiple national computer emergency response team (CERT) organizations and regulators. All are collaborating in order to reach as many affected vendors as possible before the vulnerabilities became public. Our collaborators initially included:

- The CERT Coordination Center (**CERT/CC**), the worldwide center for coordinating information about Internet security at Carnegie Mellon University. This is the first (and most well-known) CERT.
- The Cybersecurity and Infrastructure Security Agency (**CISA**), part of the Department of Homeland Security (**DHS**)

CERT groups focus on ways to identify and mitigate security risks. For example, they can reach a much larger target group of potential users with blast announcements, "mass-mailings" that they broadcast to a long list of participating companies to notify them of the potential vulnerability. Once users are identified, mitigation comes into play. While the best response might be to install the original Treck patch, there are many situations in which installing the original patch is not possible. CERTs work to develop alternative approaches that can be used to minimize or effectively eliminate the risk, even if patching is not an option.

JSOF assisted with coordinating the disclosure process, including providing proof of concept scripts for some of the vulnerabilities, suggesting mitigations, and providing additional lists of users of the Treck library. We tracked these users down in a variety of creative ways, including working with partners as well as some interesting open-source intelligence collections .

Working with international groups was essential in this case for another reason. Back in the 1990s, Treck collaborated with a Japanese company named Elmic Systems. Published information suggests the two companies co-developed the TCP/IP stack,

Treck TCP/IP in the U.S., while Elmic Systems, now called Zuken Elmic, markets it as Kasago TCP/IP in Asia. This resulted in two separate branches of the TCP/IP stack devices – one managed by Treck and one managed by Elmic Systems – marketed in totally separate geographic areas, with no cooperation between them to the best of our knowledge, until we reported the vulnerabilities. We had tried to contact Zuken Elmic early on, but were not successful; they stopped replying after a short email correspondence. (We even tried to send some emails in Japanese!) Later on, CERT/CC was able to coordinate with JPCERT/CC, a Japanese national CERT organization, who is handling the follow-up with Elmic Systems and other affected companies in the Japanese/Asian supply chain. We have confirmed with Zuken Elmic that their TCP/IP stack is affected by some of the Treck vulnerabilities. Initial research shows Kasago to be in widespread use, providing the beginning of a completely different supply chain.

Teamwork: in everyone's best interest

We have been collaborating with security vendors ***CyberMDX*** (medical device cybersecurity solution) and ***Forescout*** (device visibility and control). In our lab, we worked with the Treck TCP/IP library to identify the relevant network signatures. We then shared the network signatures with CyberMDX and Forescout. Each checked their own extensive client networks for these signatures, to identify more affected devices and components.

Note that this type of collaboration in the cybersecurity community benefits all participants, and is the first of its kind to our knowledge. We expanded the range of companies and devices that can be identified, while CyberMDX and Forescout are able to already provide their clients with mitigations and visibility, even before the issue is publicized. Additional companies were involved that have asked not to be named. These collaborations proved effective even though they were begun quite late in the disclosure process. We have no doubt that future collaborations of this type can be started earlier and provide immense value for all parties.

We also developed a script that companies can run themselves, to identify Treck products in their own networks. This will not be 100% effective at this stage, but can be an efficient, effective complementary approach, since it addresses the difficulty in identifying relevant users in a cloudy supply chain trail.

The bottom line is that teamwork benefits all players.

Download

Call Us

Phone number: +972-51-2834408

Address

Hebrew University of Jerusalem,
Givat Ram Campus, Levy Building,
Jerusalem, Israel

Social

Design: SHANIKWA Development: Linklab