TinyPOS:

An analysis of a Point-of-Sale malware ecosystem

Forcepoint

Report

Robert Neumann
Sr. Security Researcher
Oct. 2020, Rev. 3
Public

forcepoint.com

Table of Contents

The Power of 4K

What is the very first thing that comes to mind when we see “4K”? For the casual, tech-savvy person it’s probably the ultra-high definition of recent years’ TV models. For anyone who has been using a computer since the dawn of the PC era, it might mean a ‘demo’ program: a small piece of low-level code that creates outstanding (for the system it runs on, at least) 3D visuals on screen. For the less tech savvy one, it might just mean good old cash. Curiously, both of these last two definitions can be applied to our latest investigation.

When we discovered UDPoS last year, we had mixed views on both it and recent Point-of-Sale (POS) malware in general. On one hand we didn’t really expect anything new in terms of POS malware, on the other, we could easily identify the underlying reasons why these threats are still popular.

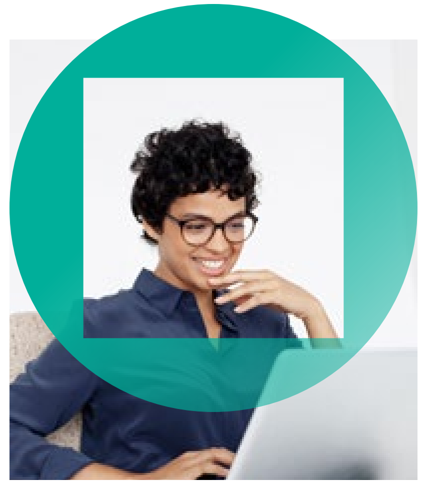
UDPoS was a warning sign that the POS portion of the threat landscape is still something to keep an eye on. Meanwhile, events such as Kroll Cyber Security revealing PinkKite – a new POS based threat, although with little to no tangible information available - and activities on the infamous Joker’s Stash site made us dig deeper.

Tiny + Loader

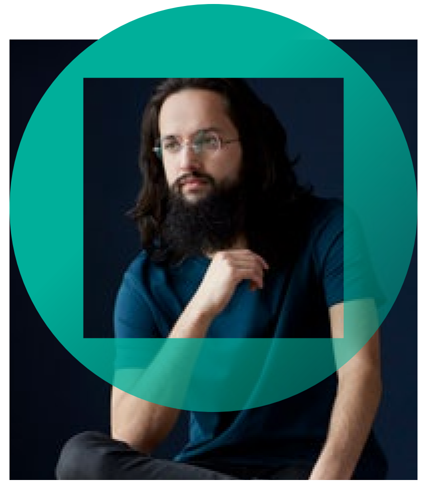
For more than a year we have been tracking a delivery mechanism built around very small components created purely in assembly. It’s not entirely uncommon to see malware code created in such a way, however trends of recent years have been towards the usage and adoption of high-level languages (C/C++/C#/Delphi/Java/GO etc) rather than low-level coding. Generally, with high bandwidth internet connections and powerful desktops there is little benefit in putting extra effort into hand crafting assembly: the difference in both delivery and execution time won’t generally be noticeable. Still, it was evident that someone still has a taste for more traditional methods of writing malware.

Delivery

During our investigation we haven’t come across samples being delivered as the primary payload (or email attachment) of an attack. Instead, they’ve been delivered as an additional payload alongside well-known banking trojan families such as Emotet or IcedID. Long term tracking provided us the opportunity to gather various samples and components of the Tiny ecosystem, as a result we have now analyzed over 2000 unique samples. These components fall into four main categories: ‘loaders’, ‘mappers’, ‘cleaners’ and ‘scrapers’, with loaders reflecting about 95% of the total population and all of the components falling into the 2-7kb range. We will elaborate more on the function of the different components later.



Forcepoint X-Labs has been collecting samples of Point-of-Sale (POS) malware that stood out for their hand crafted nature, were written in assembly code and were very small in size (2-7kB).



How it works

The loader

It all starts with the delivery of a small loader called TinyLoader, an obfuscated executable with simple – yet powerful – downloader functionality. Upon execution, it will first brute force its own decryption key (a 32- bit value, meaning this takes a fraction of second on modern PCs) before using this to decrypt the main program code.

The core functionality of the decrypted code is communication with a set of hardcoded C2 servers by IP and port. If the C2 is active, it will provide what is effectively a piece of shellcode, encrypted by another 32-bit constant. This shellcode is not 'fire and forget': it instead sees the loader establish a semi-interactive two-way communication with the C2.

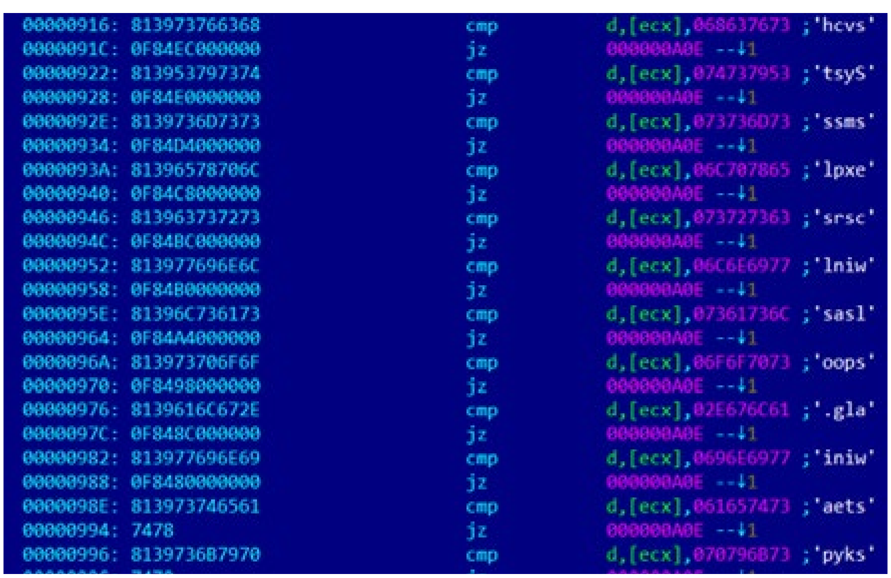
Note that the earliest traits and mentions of TinyLoader go back to as far as 2015.

The shellcode

The shellcode's first action is to download snippets of a longer piece of code into memory in multiple steps, concatenate them, and execute the resulting program once the code is complete.

The first payload built this way enumerates through the process list of the victim's PC, flagging every process which is not included in a hardcoded blacklist. This blacklist consists mainly of system process names stored in a shortened four-character long format. The list of processes is then sent back to the C2 and the loader begins building another piece of in-memory shellcode.

This second piece of code is an HTTP-based downloader that awaits further parameters that specify what to download, from where, and where to save it on the local system. Often the executable downloaded here is another TinyLoader pointing to yet another IP/port combination. This method seems to serve no specific purpose other than making the execution chain longer and potentially breaking dynamic analysis environments. If the victim's PC meets the desired criteria – for example it's a POS system – then another payload will be delivered and executed.



```
00000916: 813973766368    cmp    d,[ecx],068637673 ;'hcvs'
0000091C: 0F84EC000000    jz     000000A0E --↓1
00000922: 813953797374    cmp    d,[ecx],074737953 ;'tsyS'
00000928: 0F84E0000000    jz     000000A0E --↓1
0000092E: 8139736D7373    cmp    d,[ecx],073736D73 ;'ssms'
00000934: 0F84D4000000    jz     000000A0E --↓1
0000093A: 81396578706C    cmp    d,[ecx],06C707865 ;'lpxe'
00000940: 0F84C8000000    jz     000000A0E --↓1
00000946: 813963737273    cmp    d,[ecx],073727363 ;'srsc'
0000094C: 0F84BC000000    jz     000000A0E --↓1
00000952: 813977696E6C    cmp    d,[ecx],06C6E6977 ;'lniw'
00000958: 0F84B0000000    jz     000000A0E --↓1
0000095E: 81396C736173    cmp    d,[ecx],07361736C ;'sasl'
00000964: 0F84A4000000    jz     000000A0E --↓1
0000096A: 813973706F6F    cmp    d,[ecx],06F6F7073 ;'oops'
00000970: 0F8498000000    jz     000000A0E --↓1
00000976: 8139616C672E    cmp    d,[ecx],02E676C61 ;'.gla'
0000097C: 0F848C000000    jz     000000A0E --↓1
00000982: 813977696E69    cmp    d,[ecx],0696E6977 ;'iniw'
00000988: 0F8480000000    jz     000000A0E --↓1
0000098E: 813973746561    cmp    d,[ecx],061657473 ;'aets'
00000994: 7478            jz     000000A0E --↓1
00000996: 81397368 7970   cmp    d,[ecx],070796873 ;'pyks'
```

Figure 1: Example of the process blacklist embedded into shellcode

The POS payload - scrapers

Code-wise the POS component is very similar to the loader, except there is no additional encryption, as whenever it is delivered the operators are almost certain - due to the pre-filtering above - that a valuable target has been identified.
This component works like any other POS memory scraper: opening processes based on either a predefined black or whitelist of process names, creating a new thread for each matching one and scanning their full memory range for Track 1 and Track 2 credit card data. If such data is found, first it will be verified by the Luhn algorithm for integrity, then it will be encrypted by a pre-defined key (another 32 or 64-bit value stored in the POS binary itself) and either sent to yet another C2 identified, again, by IP/port combination or it will be saved locally.

Note the process name – the one data was stolen from - added at the end of the record.

Example of Track 1 data:
%B4XXXXXXXXXXXXXX2^DOE/JOHN^1305101000000001000000003000000?

Example of Track 2 data:
;5XXXXXXXXXXXXXX2=1103101000000300001?

Example of stolen Track 2 data before encryption:
4000000000000002=19011010000012300000 *test_pos.exe

The mappers

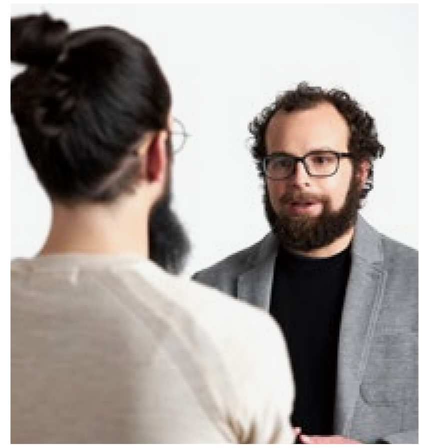
There is a special component type that we have chosen to call 'mappers'. Their main purpose is to gather information about the PC and the environment it was executed on. That's done by making a map of active processes (either all running processes or just those from a pre-defined black or whitelist, depending on the sample) and looking for local system (i.e. Autorun registry keys, Image path for active processes) and network related information. Processes are flagged differently depending on whether they could be opened or not.

For the network reconnaissance capability, the execution of the 'net view' command was implemented in a non-standard way by utilizing named pipes instead of simply executing a command shell. We believe 'mappers' help the operators gather extensive knowledge of different POS system layouts and also to deploy campaigns targeting only specific retailers.

```
"accupo","active","adres3","afr38.","afr8.e","agilpd","aldelo","alohae","alohat","atm.ex","atxexe","b2clic","back o"
"backof","barque","bbacku","bo.exe","boagen","boeft.","bopos.","bosrv.","bosrve","boutil","brain.","brstdv","cajare"
"campus","cardre","cardwo","ccprin","ccs.ex","ccv_se","centra","cicipo","client","cmacic","cmcage","concor","counte"
"cpsvcs","cre200","credit","dbstps","ddcdsr","drvcom","dscc.e","dsicar","dsicon","dsihea","dsimer","dsinas","dsipdc"
"dsipor","dsirap","dsirbs","dspos3","dvd ma","dwnam_","e7.exe","e8back","easipo","edc.ex","edcsvr","eft.ex","eftpos"
"eftsvr","egenui","epayad","epixpo","epos.e","eps_ca","eps_ge","epseng","erpm_p","esmart","esprun","expres","fastba"
"fbserv","fin.ex","focus.","focus_","fpos.e","freewa","frontb","gcomcl","generi","ghserv","gpdire","gtb.ex","gtb2.e"
"host2.","hyperc","iastor","iber.e","iberqs","icarz.","inetcc","infini","ingeni","intuit","invent","ipmgui","issdeb"
"isspos","java.e","javaw.","jpos.e","jregil","keypay","keystr","kpsvc.","liaiso","maximu","mdshtt","mdtran","menulo"
"mercsv","mercur","micro$","millen","mitels","msrwed","mxslip","mysqld","naviga","ncrloa","nextge","nmep_c","nthost"
"omnipo","opos.e","ops.ex","paladi","pccw.e","pccwin","pcvpos","pdv.ex","pdvnfc","pgterm","pixela","pixelp","plwins"
"pops.e","pos vi","pos.ex","pos210","pos5ma","posdps","posisy","posiw.","posrep","poster","postoo","poswin","powerp"
"prowin","ptoven","pxpp.e","qbcfmo","qbidps","qbmsgm","qbpos.","qbposd","qbw32p","refpos","regist","resdbs","rmaler"
"rmccwi","rmpos.","rpro8.","rpro9.","rs232m","rw5mai","sistem","sitcie","sitddt","sitdis","sitred","sitsaf","slsysm"
"smartp","soposu","spcwin","spgage","sqlser","stserv","sympho","syspdv","teckey","top_po","tpe_53","tpg_se","tpgser"
"trader","tripos","upos.o","utg2.e","utg2sv","uvsh.e","versic","versig","vitech","vmcli_","vmsrv.","vposte","w3wp.e"
"washse","wc_cor","webpos","winpos","winvqp","workst","wposip","wpro.e","xcharg","xchrgs","xtouch"
```

Figure 2: Example of the process blacklist embedded into shellcode

Some of the mapping samples contain no less than 200+ unique POS process names in their embedded whitelist. To put that into perspective, a merchant will typically buy one model of POS terminal to use across their business, with these solutions generally covering both hardware and software components. As there are no trial editions of POS software bundles freely available on the internet, gathering insights of the components is no easy job. Judging by the sheer numbers, it's safe to assume that many years of background research and information gathering is reflected in these whitelists.



The cleaners

As their name suggests, these components are responsible for cleaning up leftover content once the operation was finished. Such content can be running processes, Autorun registry keys, scheduled tasks and files in specific folders on the filesystem. Just like mappers, these modules are also customized for specific needs and won't always include every type of cleanup. Some would only kill a specific process and delete the corresponding executable while others might do all and also send the report of the successful – or failed - cleaning back to a C2.

Obfuscation and anti-analysis techniques

While the individual components of the Tiny ecosystem can hardly be considered challenging from an analysis standpoint, they still utilize some basic obfuscation and anti-debug tricks. Note that not all of the techniques are utilized by every component, the early stage downloaders were meant to be protected more. Some techniques which were identified are as follows:

- → Call / Push obfuscation
- → Fake API calls
- → Junk instructions
- → Imports by hash
- → Icon resource randomization
- → VS_VERSION_INFO randomization
- → Dynamic loading of all imported API functions
- → Simple encryption layer added after compilation

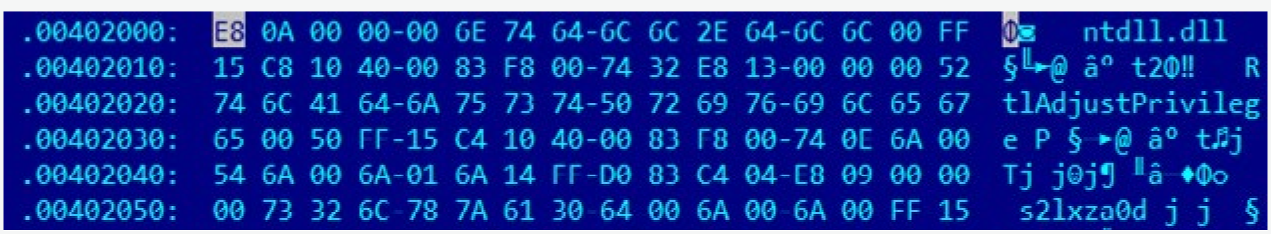


Figure 3: Example of Call / Push obfuscation

The infrastructure

There are more than a dozen IP addresses utilized by the operators, however most used in older campaigns are either out of service or have been recycled by now. Even though the ports used to change frequently (sometimes within a day) the IPs for ongoing campaigns are static. There are certain port numbers reserved for specific activities; for example port 17771 is always used by the memory scraper modules for data exfiltration. For a complete list of IPs and ports please refer to the IOC section.

IP	AS	AS NAME
5.8.18.222	202425	INT-NETWORK, SC
23.228.232.92	40676	Psychz Networks, US
31.184.234.108	44050	PIN-AS, RU
46.161.40.145	58271	VSERVER-AS, UA
62.210.36.112	12876	AS12876, FR
77.72.84.115	29073	StartUPG-NET, GB
85.93.20.42	57509	LL-INVESTMENT-LTD, BG
85.93.5.136	200998	EMGOLDEXNET-AS, DE
91.197.232.26	200363	BK-AS, IE
95.154.199.104	20860	IOMART-AS, GB
179.43.147.209	51852	PLI-AS, CH
185.174.102.20	8100	QUADRANET-GLOBAL, US
185.183.160.137	206766	INETTECH1-AS, RU
185.248.100.188	44812	IPSERVER-RU-NET Fiord, RU
188.126.77.137	42708	PORTLANE, SE
193.142.30.201	59580	BATTERFLYAIMEDIA-AS, RU
193.28.179.200	58146	SVOD-NET, CZ
194.165.16.165	48721	ADM-SERVICE-AS, RU
194.165.16.166	48721	ADM-SERVICE-AS, RU
194.165.16.199	48721	ADM-SERVICE-AS, RU
199.165.16.165	195	SDSC-AS, US

Note that some of the IP addresses overlap with IPs previously used by the Cerber ransomware. Due to the nature of how Cerber operates and the lack of additional evidence, we could not determine whether this was a coincidence or an actual link between the two different threats. The finite number of IPv4 addresses available can occasionally result in the coincidental recycling of addresses across unrelated campaigns and groups.

Why are we still facing POS-based fraud?

There are several issues that come together to make POS systems a soft (and tempting) target.

The software side

Firstly, the solutions are frequently based on old software technologies. Lots of POS applications are still based on POSReady 2009 (Windows XP based) or POSReady 7 (Windows 7 based) platforms, both of these are reaching their EOL in the upcoming months or years (see table below).

There is a new Windows 10 based POS platform called Windows IoT Enterprise, but migration and license costs can be challenging for many merchants.

PRODUCT	EXTENDED SUPPORT END DATE
Windows Embedded POSReady 2009	4/9/2019
Windows Embedded POSReady 7	10/12/2021
Windows 10 IoT Enterprise 2015 LTSB	10/14/2025
Windows 10 IoT Enterprise 2019 LTSC	1/9/2029

On top of that, POS systems often have some sort of remote access application installed for remote management and troubleshooting. This could enlarge the attack surface as lost, stolen or unchanged default credentials might provide access for the adversaries. In certain cases, if standard database engines are deployed and data is still stored without any encoding or encryption, it might also provide the ability for the memory scraper module to not only intercept current transactions but also historical ones. Several TinyPOS samples contains SQL and MySQL related names ("sqlser", "mysqld") in their process whitelist.

The hardware side

On the other side, hardware-based restrictions and outdated standards are an even greater concern. Magnetic strips (Track 1/2 type of data) are still being utilized due to the lack of EMV support world-wide. The EMV standard was originally written about 25 years ago, but adoption is much slower than expected.

The liability factor

There are numerous types of credit card fraud, with POS-based theft making up only one, shrinking piece of the whole. As total loss climbed up into the range of billions of dollars, banks were looking for a way to shift liability over to the merchants. In reality this means since 1 January 2005 (EU) and 1 October 2015 (U.S.) merchants are liable for all non EMV based transactions. As a result, adoption of EMV has increased momentum in the EU, but it is still lagging behind in other regions (mainly U.S. and Asia).

The human factor

An unfortunate side effect in the form of human (im)patience is also contributing to slower adoption of EMV. Swiping of a card is a quick and convenient way of making payments compared to entering at least four digits on a keypad. There are certain regions of the globe where people are reluctant to remember yet another PIN and don't wish to be slowed down by the process. For them, new and emerging contactless payment methods will be welcome. While some of these are still asking for a PIN for transactions over a certain limit such as PayPass (and that limit may vary per country), there are also semi-limitless options like Apple Pay.

The connectivity issue

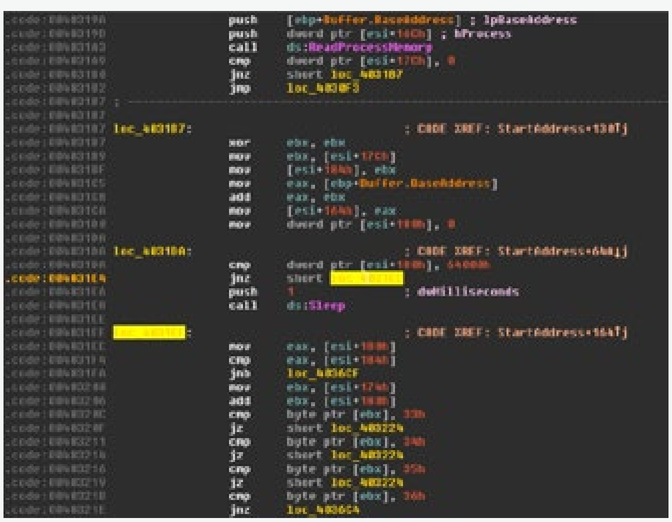
POS terminals can utilize different ways of communication for both in and outbound connections. Depending on the nature of the retailer, some might have strictly Ethernet cabling, but the likelihood of using Wi-Fi and 3/4G based communication is becoming more widespread. There are certain scenarios, i.e. taxis, where cabling is just not an option. Also, depending on the total number of terminals used and the network infrastructure, there might be servers dedicated to proxy requests towards the bank instead of every terminal having a direct connection. Whenever there is an unfiltered direct internet connection, there is a higher risk of data exfiltration going unnoticed.

All of the factors mentioned on the left, are in play when it comes to a merchant's ability – and willingness – to protect against POS based attacks.

Forks of a common codebase

One of the obvious questions is whether TinyPOS and PinkKite are really different malware families. The short answer is no, they are not different malware families. While it wouldn't take a significant amount of time for cybercriminals to recreate the assembly source code of the components by reversing existing samples, there are certain improvements to the codebase - especially to the memory scraper code – which can be found in both.

The main difference lies in the method of data exfiltration. TinyPOS variants aim for direct exfiltration over the internet, connecting to a pre-defined C2 and encrypting credit card data by a 32-bit key. The PinkKite approach is used in scenarios where direct exfiltration is not an option. Under such circumstances credit card data will be either stored on the local filesystem or the local network and encryption is done by a 64-bit key. The exfiltration process will be manual and will require persistence on the given system to access the previously saved data. As many PinkKite samples contain internal network addresses along with login credentials, it's safe to assume operators of the malware are gaining these by hacking into target systems after the initial reconnaissance stage.



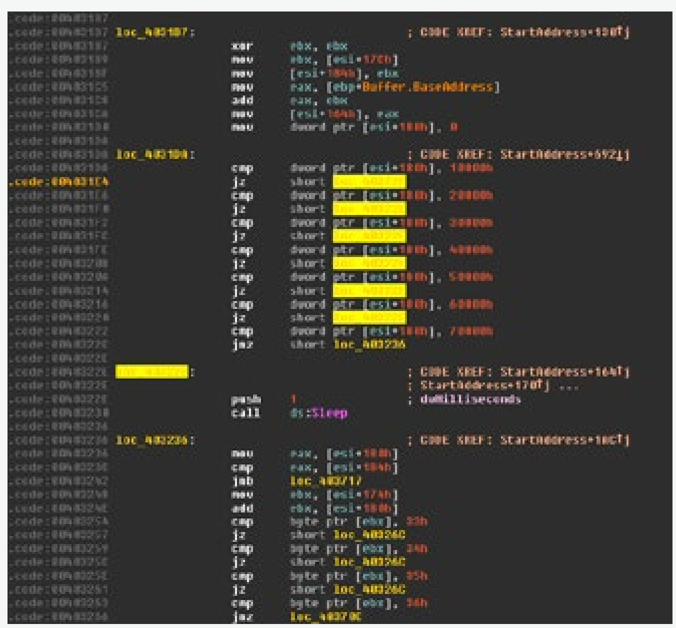


Figure 4: Example of code evolution in the memory scraper module

Protecting against POS fraud

Individuals

First of all, you should make sure that your credit or debit card is EMV compliant. If it's not then your only option is to use the magnetic strip and most of the time that also requires you to hand over your card to a 3rd party. Reaching out to the issuer of the card and negotiating a replacement – EMV-based, or one even supporting contactless payment - prior to the original expiration date is highly advised.

Merchants

The most important thing is to confirm the various connectivity options to and from the POS terminals prior to securing other aspects of the network. If the only option for terminals to reach out to the bank is through a proxy and no direct internet connection is provided, that can greatly reduce the chance of immediate data exfiltration. Also, in case maintaining and upgrading the terminals is handled by an external party, try to limit having open ports only for the duration of the scheduled sessions.

Conclusion

As we can see it doesn't take hundreds or thousands of kilobytes of code to steal valuable credit card data on a large scale. The Tiny ecosystem is built from simple yet effective components – each one being responsible for a dedicated task – written in a low-level programming language that only very few programmers (i.e. people working with embedded systems) consider their 'weapon of choice'.

While Swipe-and-Sign exists as an authentication option for card-present transactions, POS malware like TinyPOS will continue to be effective. We strongly recommend that retailers and banks aggressively pursue a faster move to EMV (at least Chip-and-Signature, preferably Chip-and-PIN).

It is also recommended that an audit be performed on any system storing and transmitting personal data in relation to how that data is managed and stored. The goal should be to make it harder for credit card data to be extracted from the retailer's systems. This includes while in transit.

Indicators of Compromise (IoC)

Mutexes

edcfix
srvrcard
carmahot
s2lxza0d
t0cnhig9
sqfinuk32
thmnhig9
hfDscs
ntcrash
xcm3264
fswinine
msoft
fswinisrchx
fswinisrchy
fswinisrch
fwnmsft
winfx1xf
joycml1
sqlsvr_0
l_smsq32
lsms_32

IPs and Ports

179.43.147.209:40071
185.174.102.20:17771
185.183.160.137:6317
185.183.160.137:8181
185.248.100.188:7454
188.126.77.137:4119
188.126.77.137:4357
188.126.77.137:4358
188.126.77.137:443
188.126.77.137:6317
188.126.77.137:8181
188.126.77.137:9090
193.142.30.201:1192
193.142.30.201:1193
193.142.30.201:17771
193.142.30.201:17799
193.142.30.201:9290
193.28.179.200:10012
193.28.179.200:27117
194.165.16.165:1444
194.165.16.165:1445
194.165.16.165:17771
194.165.16.165:19991
194.165.16.165:22143
194.165.16.165:22144
194.165.16.165:7450
194.165.16.165:7451
194.165.16.165:7453
194.165.16.165:8181
194.165.16.165:8289
194.165.16.165:9090
194.165.16.166:17771
194.165.16.166:443
194.165.16.166:444
194.165.16.199:17771
199.165.16.165:17799
23.228.232.92:1192
23.228.232.92:1195
23.228.232.92:1196
31.184.234.108:10011
31.184.234.108:10012
46.161.40.145:1192
46.161.40.145:1193
46.161.40.145:1195
46.161.40.145:1196
46.161.40.145:1393
46.161.40.145:17771
46.161.40.145:4356
46.161.40.145:4357
46.161.40.145:4358
46.161.40.145:4360
46.161.40.145:443
46.161.40.145:444
46.161.40.145:8181
46.161.40.145:9290
5.8.18.222:1191
5.8.18.222:1192
5.8.18.222:17771
62.210.36.112:27117
62.210.36.112:3341
77.72.84.115:17771
85.93.20.42:1191
85.93.20.42:1192
85.93.5.136:50011
91.197.232.26:17771
91.197.232.26:9090
95.154.199.104:27117

Files (SHA1)

Mappers

16dd2043503d8b68075362095214f7fbdbb28a13
2a21258b6954f66d8c29d59e5430b0e139fffc57
31bca61b30feaf4f3d8505ba9ffd54e2a5d1ac25
3511a6d3bb94a59dd776f66a9fcdc8115707355c
5917b72c2e3cd20990036ac5a31e7c62d041afe1
7ebc1745c5f0a6ae42bd287706f01f6db2792ccb
83c2f1bbb94139a9a6028ea2f4e6d8aa39aa1d61
a85439890a3a9538420d63a93dfb6e12ed3a5ce7
bd04b3c0840580e072806074bd309b07e61c7599
cb0e0851ace1e670a6d7826c1d05c738a1bf9e12
e8acf26abdd614a06f435337fe7a21e51893869c

Cleaners

4d181a749cea2fd82aa80f319336b08b8b961e6e
6a5eca33e5fe702ff4fe3eb47a7b42c85b1473ae
7719cd1cf007906e22b1028fe5ee65440eec3d33
860060d72ce8f0a1030110586d7b5db6d2bb8036
8eea84d8926dce44663ae1993f2623af3f40ab3b
b07baff55d9a43d93246583d34f62db7127787ca
d0af2c082d8e6ca707e4fb768715dc8974ca62be
e8fee73aaafd8b9e75441c9767ab50a7ce5f9c31

Files (SHA1)

Memory scrapers

00a46a475d56b0e56e0522d6736330935aa64984
051286924a39d9c100c131f7d48600d20d465cb7
115eaef370396843a9e28215b7267ec9559f45f3
15e8c23c989ebf7df86f831b4c400abdde9f631b
1ad1019b1463216bde562ab1a60c877b8da4d7f5
1b7fc7aaaca65c0c12222ac51e7f02f198266b07
1d9e2bb347b6d0a0b341926a603c8f0daf25bf09
1de5d87aa3f4410a18157ce2bf8d37685e908798
1f98c55b57036ffac6fa08c0cef3cfe54a5a6dad
23f3dfe9d07e82da22f7f597ae59a20f3e0bb0a2
268f6d661c5c51053161768cbadf38930b56cf09
26bdfe45a89956686cae31f9c02d1cd54de486b1
291eae72a8e461fb3165be87f75e51fae7384b1f
2ba124738f1bcf27bb2a598307309e7485cccb46
2d1d69a09985b0572a1bfcb04f1a647370592511
32837616c06390831aa9a73f314819bacfa94db9
342f66035ed3485a3941992b7538410928cafbf3
39273383aaf7e8e355f81fd8318a4b0306fc9573
41c9f1e2853e550ec2bb64efc40a553abec0365e
44854f7a8f4ae2c2c46d8a008b0988e515bc588d
4d17bdb94e2999cdf8f81d2689489d4269896ff4
5021a421a93d0550b89cf3d547232b34bf1b4b93
523923a402e468185614888ae65e1dd7df314e47
531f5cd44f6363096aa162bb4edc1a0a4b6f3bf6
537df154deebafdc269d056994fde2100feec1c1
5725bce50fb1dfb92b67eb831a0ac4fb41df2cba
58a6a7da1e7d9fc96773681c33ea8634b212b178
5bed33f29ee823c742100dabe5754fbe32521ea8
5c4ec897c48dfbfe5282ce514ab8380734d2cb67
5e7d26565f0131b285404065caf600f8464a45b5
66a008091f01824b72c4619ec75fddc9064bf695
67db6a595d3c8c93a996d9cc5cf309f286d7b617
6afbd9981837b43866dd89235081b0a138f8ceea
6c28d0ebe7ca0dedaa6b67564f5b9d5493927d6a
6e1487e033c5a159c1a0a7a43609bc5856daebe3
72861b46616b5bbb9fc11177e756787014f610b4
75e0d61c8727f038ae15f92aaa73698463fabf50
7a4034c9442237a93e60449c67147b48ae4b3bea
7ba56e784a37f0f9a41cff839d53d4e3ac0318d9
7bf0e15930f5733f276207f826d37a351824b75a
7e5344b14d24291f43c19f6acb1eac68899fb6bd
7edbaac22f3994cca2d93dafc4a2b4ceb8a68369
858d80f99c57a097a8d3b6af06abce058cf724cd
875f7cd53209d38552bd1bfa769a633439bd2db5
89928295ff61ddffa9fc9c709d7556bd0ed4a9da
8b72238eb623c8a16ab2968a2300246046d86928
8bae12e24a4424062697c2c31ef0bb20a9bc0e7d
8c6c1dfbcae959a735586b8f5b1a2fac5323ef59
9adad458941038d359f05f4f008ce3bc586ef9ba
9f2f7e4d4d036b4b566d28ebd69d95fb7c8dc2d1
a5f166e03dfa09cd3460aba4bb6292def4444341
a8a868564eefb7dac76d5a595adab257ae6910bb
ab5f64f43693e51ca7316e28fbd902d4849dd025
abd708177a3b1a34a5290312bd0dac7ed2257d36
acbab2110cd7749f4f4febbfebd38ab72fcc7c9b
b03b7b641029c445e299ac91e87ca92905527540
b61637f78fdae3b4ad7882c367976f773e9e59ed
b99d743dcce352b1fa5aa2f845cd88f248199ecd
bce7a1d51239ee26137fc39a9f01ab2c5ed0b5df
bddeae58211c4ea524db367ff84e550921cb4bea
c4bbf71539e71b4171d669c16b3d17b20fe820c5
c6858a96d8d73e563ed2864ca5bb1327291ca6b6
c9f169da716e081c5a492ac2244f475ff652991c
d5ac74d5f581b4bf2e45311fb94a07198de33ce7
d8ad68c12e5f813e007ef1f4bb12b9fbc930ddde
dd587590091d523912d80ca0c6855423aab26126
e1478e82094c09c9765658e2faaa51ff0f0f8a73
e255ca839fc70785b13054337bad6f41b1f65fd8
e44c25157c4c65fc6ade179d4d3fa0407a38f880
e4fa892f2252e562001b28b1c6f407f80832db62
ebecba19e7fd6f80c3cce77288c38e6c61bf7446
f03db920746807466419c4a5096496afbf8f53e2

Downloaders

Note: for brevity we have provided a reduced list of hashes for the downloaders. If you wish to have visibility of the complete list please reach out to us via https://twitter.com/forcepointlabs

028742c5c6b3deefde295ed9ecf95fa98c04dd4c
02968031a3afbd3447449e118f6e89f49853e5cf
031c3db8e0490a4379571bae2845742897b744b4
048252f13ec38b88df0ac00a3afef6f3226e0718
0486218f663b18db97d47ac5c10cbe16caf29ffe
0626ab3fe9156b44e07b9c4c1a86516021ec4ae8
09689a26b45f5a7339d327557f3827ac229d8c20
0a76fa8dac5ff5d6420cf1c75f2823e502c3ecf0
0ab364bd8b6a3b88594012c130fd473bdb3ee6ef
0c28d29d57ff72df7b3ad1042d14dcb9a1c16d90
0e30d6dfec2b296c5d40baab2f88266e239cc4bc
0e7cef887e54962ee4d5dd810b95030b7623b870
0f438238665eb82b756acf1ec7e566b7909b4960
109f9e62014e14c5c9605c1f01d0603e26a915bd
1106c05734faf7c316bc3cd766664d54b87f604c
121a259fdce337ea7cb1e7499a0ce7a797721c5c
126b62a33bc8598f0e4c58c3b2bc657e2b3375b2
132cf9d6db8dd64786568d445ce8b650d6f406a7
13834ad9ac1833c83521755b09074b38f9033236
16634b627e38c076a5423d2164b1a594587f9a9c
17722edacb2d367391eb41d09d79e3356d225cf7
17a8ea843d8e0fb929e637d3e6f1ad26b8982347
1820082cdcc7665d07a169ba33d0dd6105bde422
1982eaf61effd0229f272846a7728406482ba3bf
1a445c92b2bfd5480525c075ec577609d1af7ca1
1ce7c92db916b73e3182f17670b7d550cb76718c
1d5ce7d15643646a28ab148691c18047cf8fb9ee
1e63aaaa4eb4ba61b48caa642e7743745ff81c89
1ecb9b3124cc4b51afda45fe732ef1243b39a8c9
1f4c0ac87218c12cfd02e856a8e7a61027dd5e10
207ff0f630e430d25c788ea6d4ea3bbacfeb33d9
20a4820dd61e5d3115630162ac19f630a688198b
215ed11523ea6a4d918acb09559823a337b62834
2177ba22e9724f7826ff8511663e28a1c29232c2
25cf7e8c92fb13a96108aaa596adb228117f8765
26389a6904b29dbbb90e20825eaec7e2f55b59b4
276660481dc2655e8b15cbd4de2007866e77b463
282b6e6cff447ccd0525e9074f8b68f8060c640e
28522dab2b86976460d60891eae13a8013a2eb5a
29bd8b56bcdb53b13a264bdcd76150ff41756645
2afef3be6fb9348ce4876f5d1d726b90685d5a91
2b8f7f7b6af03dcdead6430e26c3dcd3206be497
2bb5236ce95c1e8db044520866532c559c685cac
2c0fa9ba6152e0e6ea5f80657bfe1898c021b633
2f486bf55b43aa39e0a64ec0295d715232a85607
2f791f72e4965454c3e4bdeecb9c22caf0e5c14a
322e95fcff92ac692c32367caccb89bffbf3556b
3500a97643488ffb5d21339f621b85169b0c7fad
35676293efca634603cf3e3ef65961ff92e78199
35c68b24cb9118362233a9cd9071e78071f40b1f
35f4a8d1574b3e86fea210f15d1961f4609e8835
367df6425eb316f98a10f6ca112fe65fded46771
383c047e591e1d78ff392e1aa8b8fbedd8a9bbaa
396c93b9d3c5819b0cae2a9c6e33c2f61e6e095d
399b17b3228a9c91e3752b643c5f043807002865
39c2b5c090308cb685fc0a93c9faacca180282e5
39d2e84f372a89a07bb84868b8b5da3b00ebc6ce
3a6d174040d43f7b874043d37aca81087b0bcf80
3af3e9e3b20cbe50611aba51fe8c451ec2b61937
3baad1b48006826cf135eee51eab0dc793d5d35a
3ca213bb8c0ccf163708888eb47f22cb58dc8466
3f3814a5a235b597d79070df6a4a746422fbbe39
3f64a3910ed8a4e33c115438ed455d9de5ffde40
3fe5815f51ed1b093c1a1c274420123298307c66
4176bc6932eac200c079f0d4bebbcb469344b301
440eae6bcfe4830f31531e61dc309ebd30ca0df7
4462dc30538a9d12c84c6b3b014dea1a3811370e
44fb9ad80107e143bc14b62f367fde40075e8a99
45e240ace247c55010ba75c3bcfa768884d83e46
45fcdb1d97efb3786d6c3e0bf50fa700aa423e3e
46e86b75ece45cb8e9176a7b8b7213b27d0e3ddd
46ff0cca7f6f381d4065280451a37d45287a4442
483bfe2af5ea94d98f050f8b768f3cd9f207fafa
4857eee40e0e14cb3e5173e37e1f728386caf947
497b114ab46868bd35e969fcff2ab92a639578d9
498c12a2c00fbdf22954d9b06737ae4e5a48c68b
4cc3438cadab604b3df3a25a129a1b354b6aaa25
4e829b18cbc08a03a08812c15385dce479f4fb91
503ed181b3f1015cb18adf2687e36d9a4ba00a95
50a03cbac2c99b2549e178ddfa7a4a9f6dbd89ec
524272ff65275f9a79376446b8d984cebd76c559
530c31c9e389d0573498a6d5e95d1f221ae84199
565704b49f7a60c92c5830039f0116fff48802b9
56ae1c8692e21ac9fece691b0e502667b649b187
56f4a37fecb47aecb27c46ff5e5f7a26bd0e4cc6
5739738ab6180c11343a55356a1c3600430ad3f7
586ebce580be8386cf2c755ace71f4e5d92cfac9
591201f141e82a78245e46e4c54a5cdd531a8252
59a6a19eb128e9fdcb90bdfeb6bffdc2798e1c8c
5a45ca9d44980dd32b7cd7dd08c4f6faa33caeee
5a469fbc9c235c6fe2460e811f00db4e93eebc72
5a51240b178631909bb82900f09121013f72c0e3
5afaa637a7451fa49394942d6680c92c363e8b95
5b97df8ec4f0fa24287a70f487be974e4905a118
5e6b1e562f93111a140ce56933c708640104bd37
5eb5d3cd7861eb5dcd71732a20ee1b151f23d225
5fb2d5015d56837adb2e899c264e86905f26da45
6053ab3ba74314f5c8fe6be8ed254ef8edbaf725

63131ea6923a5b0db809937786b18a82e3ed9e29
664d3ee0df4e213579562f38bf19e781ec5fefc5
667ab38c032f797c29a107e421157514aa7a94cc
6741386a9ee83ccf5cd169e0200b60d0677a05b7
676235d4ed5c7083f657afa10b82a81c2e1856fe
67f2c8d5d53070d521f92e7a757fd5be8a0cabea
691d5924c02002fef954dfc2fead1ff50f8165a2
6ebdc1e92b7dd5e28c100e425acd9d23f7232757
6f6a741d12774655b5bb5f1833be8dae5e170b19
706f9455df5211a55e25ae9cf982c79fad6f1f90
7086e191d7c478fb679820bbadce2e08898b656e
710e917f28e5fd4170b8bf58b48b26329fc1556c
7395f1affe5a062f211364b931b6778cdfb43bb4
7439ee6de5e2497f6db727a7d1d80bc36da7ad71
747d01d72f06cd29633368b5dc0c35eea0ccf3d2
74e6dc09a0761f9d4475cd9af1663e4253f7d953
7538d911be0ecd29def2cdec01b04df64dc4991c
762bd51d78656f8a4aa50293c022fa820f81e061
76564900336d348795fc42d9d1730b3a7257c676
783dd0ad9a4529ba4fa0f766ea08de31409dfdcf
784679bf4c7bae78c1beac61a227b7cc59d4a87a
7b512322846addd41679800f830ae32c62cd9e32
7d9cbd2cb52af63ab432f178080089696316a5c7
805159f8a46dde2eb8bb257c0b5bf50711d06a2c
807f31118ec53683a7979a322ae56d9839314903
8104a7829603615c588107de341c698328a5baf6
810bc5db51ee4d91d8e84dab97dad56875bdb91b
814dd9a259426f8f31b8814689bacc0bb67702f2
818eb0eed3b4a2ddcbe267b3f3f9c7973f042f8d
82549c8968be98ef8e675245ef40b464f1dd6078
83a0f1c40911d972d79173f87aeabbeb5429eff4
8525776fca90791e1d917fdee3c5735da79b8af0
854c49377d06b247f376375b55b03d62e33d3064
87facc604f5733f11f4a7951e837c41b43283d1e
88369ffe15fc13a93cebd6aae7466609a1ef3286
8a44828238d05cb4ef58221afafd0cc08cd150b3
8a757da747a782ced428ee81e53e6326f1ab277d
8bb29c40e6811394bae35a0d3ce0c424491371d9
8e9d20e76222a48ec1ec7d272dd9e0405308617e
8fdae882a2e94d6bc7e1a34211af2efdce196626
916ab802543f2d94fce1b10ddf394f213cea0709
92a3839c80a8772567a5d7089c178cdf3fecada1
9382fdc5b859aa94bff13dc43f4a8badaa888fee
93adbf7953313dbf77bbbf31d304a6febff18767
93b6b210e8ba8544aed0cce3f91a6e9458d440cd
93eb76e783245d618f91426303c6080606645a4c
9537ee5d77c5d3f2a605d4a398df43ce6bc013ea
957645e5ea6d97a4383672021a7a74887b725817
9674bdb8ba0f9b8e7473b2f5a6396b306d90c921
96ee1b6b95d5511aa03ba78e497c18195197ccba
977eb89ca20f1fdd0976bf676f35b85f5c5d2c9b
97f6069b53a44518803455d4007360950716a467
982291053af978431e90c8e1ef873c54d141de3c
99503ff18a17d318b7d27d16245fee4bd737bbef
9a111487544128c76b1516d8090d5e5a94dbb44a
9c4fd56650e47900bb5036a338a73208f3274fb4
9c736a0d2da4822d712db5f1c9858b4cbd28d19b
a0338aa0989eb9977e96c8d81a6a263c53aeaf93
a0907b7ffa6782f556b17c17e717be3a42b45975
a09cd885de4639d3c9d07b818f63e646b7d880b0
a1ac7eadab3bbd44d23db62d488dba006f26b558
a362f72d8d784ee78b2676611c872df62897ecc3
a50803d73420ea82700d348e72168c30f0df14c3
a541b8ec349d1ed0b98c995618a8eac93d097fc6
a7c622d8e973551d222539864082cac8d25ecae0
a7cda70d88d1b15ed1f4acec3b86dbecda246fd4
a8920727e71e4aba574a74bcf6374c97af693ad1
a94eae38b37ac80722eccffe8724933b951d2b58
a9d272a9f018ec9c670c19d40a2bc2aa617207a9
aa447034ec546e54a1b8ebb482377aae5e0dce17
acc6536d222bfc1cacdf236b16268d1028d82fd5
adbcbfef86df52099542d01a45db9b8037d4ec69
aff7a1370b9550983afc726a994a262ba25bda91
b15f97a4fad3bfacb43334dc720c7a60227653b7
b230b185c3c39c2c20d90fc144e20b73c6bf1aab
b237e778d3bd18b4581fb380a9961e6677969290
b2fa05411fe2038e4da1880144f94c593cf1280a
b41b3f714cf82adc791d79b72316063cecd01e89
b44dc697bac73eba1febf3dab1c946831fcc367a
b5adc5665079f40e60b4941ace20ec32345963d6
b6bb753c97d0d151af0e56d02b05bc3f8c420517
b95d8584ad27f9de6cad837733bd4d723b0a4efe
ba2362271abb5831b3e529aa80138c132e8b5ba2
ba4eb346f9e82e0dde7eef501ab88565f5591da3
be73c53a04a37643f63e8866201466a606a96a3c
be8c1ad0d4d97d73b04afaa27268bf9b1a44f99c
c184eb1afdc919f25432470c99a0b0283ccfa35b
c276cab7bcced7f9b5f76ba957251df2751b36e5
c2810c51e2be320d639fa5b1cb295805c8ea4ce4
c3bd59e6cb6eb4d4bcfe95d13755dde8bf0335ec
c45deda16af0a9b0ce160c80909ef6dc6413d2d5
c4bdc0e0f2a3d9ee77d7d9c37e95572e78f29cbf
c4f9eeaaab4af36272ef7dadfd9093c653ad2b6a
c5b2c38f6a6093602a490f32dd697e1234eaaeaa
c5c790b9680e1b39765aaa510ecc527534d16645
c5ed0d681f90ee5f60a16116194caaffd60ab012
c7200012bf32e426081c1f9e8a8167fa88e2fac0
c7b4f93fd4ab48b5fd7df8b4b6f4ebe4e335dbdd

c9026001509ec0e8b899ba92bc2b2e36b4c5f301
c93f56fccf083f6e711601d7b6d215c9ceec53f9
ca2ceaa05773674738e7f62715afc5210dbfa4c7
cac8c19e85e2031ab0a69a598f170aab7ed13f38
ceb06d3aeab91ce9e0a8989759d05e48a054ff9d
d10855e6a2f3f3a68f8aa2c71180da0403e23350
d1d7e0dfbc2105abf14a14955825c2ea70fbe1b8
d48ba6dae4be84be6a202c60c2443cf8c66adace
d498e1c8aea472e8189975b6654e42bb38cd113b
d506744278898ff0363cebc435b7e5674eb84a8c
d511f4f6bd39755195c1e5949269633960e2c85a
d646785cc5d4e3558a83d5b5f960e9076fb8b494
d6d4e97af00893de62257162e30b0bb20790b6d6
d7526453a3c1c870272430e6e6376dbe09f9cca3
d7f8a1313916fc49992ace33b3bdefdf1f309c11
d8c3ea83c90fa53d07f1847e5047596b0a484b16
d91c48ca75b606f00001bbe92372b23c9de39f99
dca5fe92be8953f56955984f416562387d74f88f
dd3886b72843ecef25f0ee83b8f74ace3da391c1
df67a164fdc6581ed1c226f5ad4daab2deb3ba31
e0297ce574710392f6938bd9f5395128632a56f1
e03facf2248681cc69278fb384693fc707d18101
e0aab102726ae49c97221a1a08ec83b18ee94d9b
e193e4a8fc051dfc596a1ad70ee1ab99a28679bd
e2fbd0b4751145b2d44b15741a3f935c545e06aa
e38734e2f8f7c7f0ecc9b62b6367921f4944bbfd
e3965b57f7d372a01e23c7a9aa950cc557a57a42
e66f77068143bd3ffc4f5fd8cc33dfa77df85a24
e6f62825321f537a4c41817404ce8fb284ebd836
e7cd69a140c7733f3be9f43d25bbfc78bff94d97
e7d3d786871c8d8bf07df58bbf1a86000179ce0d
e82330bab3b5041ed0db605bd6c7b04229e98c1f
e9b80754e66a6ddb1843f9f57e6e530be6436554
ea396fc739f28b336a443847462739bd21c94270
eca1175c6c6b06aa466870c51d07326ba07b5bfe
ece6cc313dc9a0f77c840670c3dc3ce5f33c1d3c
edc5c173fbdd8f12c24102b01286cbd90abb5ee8
ee4982482b94661cd2b50abe9ddbfaabc3bf1f18
ee7aadcbec95e19c6d9ab5ef2beb3ce76aac19b5
eebb9cc2b1394b827255203f02ce772591a61956
eee87504008b8e2c0d287c9e3143ee321ed068a2
ef4381a240b7455c7f532dc7bbb8938704d7abd9
ef5ab0cab28b0700f1dc3e243867ae1602af808f
efc042d4501f92360e9e301cd3fc5239deb13ebb
f06dab3f2b17442989d9f48d4064c2adb73b8f06
f0ab8f0d60e4faa56df94253dc5a2cd6914f4741
f23450d854d42378a3fc9efef744006d8f710d25
f3624295cac6160378adc3c3623f5522f1756cc3
f79bfdaa913e974754062fdab85c4c30b5034717
f9219e3215d94b82fc89fe380d48f451ee8d844e
f9da3b527fdd23823cd89b28bda9b42090965d82
fa613fa2734c4e782baa532a5acf5d4b2f2d4c28
fc7327a4030da926eebf778101f8e24b5d3d3681
feb597402d2de8e50b8e05a9ea67ead1dc69d69a

About the Author

Robert Neumann is a Senior Security Researcher in Forcepoint X-Labs. He focuses on various short- and long-term research projects, ranging from small scale malicious campaigns through niche malware and file formats to in-depth investigations and threat actor attribution.

Forcepoint

forcepoint.com/contact

About Forcepoint

Forcepoint is the leading user and data protection cybersecurity company, entrusted to safeguard organizations while driving digital transformation and growth. Forcepoint's humanly-attuned solutions adapt in real-time to how people interact with data, providing secure access while enabling employees to create value. Based in Austin, Texas, Forcepoint creates safe, trusted environments for thousands of customers worldwide.

[TinyPOS-Analysis-Malware-Report-EN] 26Jan2021