

EMET 4.0 PKI MITIGATION

Neil Sikka
DefCon 21

ABOUT ME

- Security Engineer on MSRC (Microsoft Security Response Center)
 - I look at 0Days
 - EMET Developer
- I enjoy doing security research on my free time too:
<http://neilscomputerblog.blogspot.com/>
- Twitter:
@neilsikka





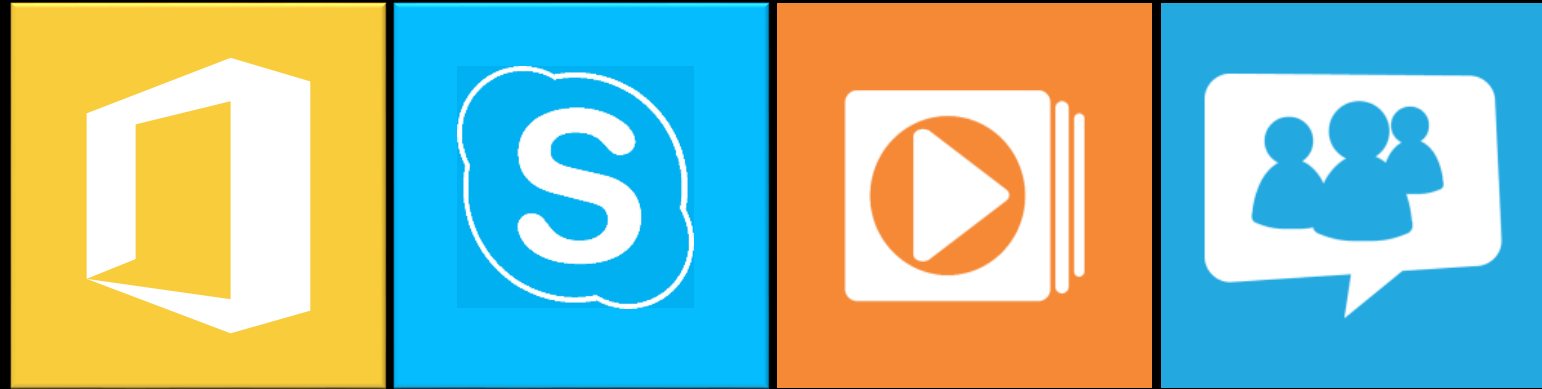
OVERVIEW

1. What Is EMET?
2. New Features in EMET 4.0
3. EMET Architecture
4. PKI Feature In Depth
5. PKI Demo

WHAT IS EMET?

- Mitigates various exploitation techniques
- Not signature based—behavior based
 - Things like stopping shellcode from reading Export Address Table etc
- DLLs dynamically loaded at runtime
- No application recompiling/redeploying necessary
- Can help mitigate 0Days
- Works on all supported Windows Platforms on x86/amd64
- Giving back to the security community
- Its Free

COMPATIBLE APPLICATIONS



CHANGES BETWEEN EMET 3.0/4.0

- We added Certificate Trust (PKI) Mitigations 😊
 - Our first non memory corruption mitigation
- ROP Mitigation
- Some ROP Hardening (Deep Hooks, Antidetours, Banned Functions)
- New GUI

EXPLOIT MITIGATIONS

- DEP
 - Call SetProcessDEPPolicy
- HeapSpray
 - Reserve locations used by heap sprays
- Mandatory ASLR
 - Reserve module preferred base address, causing loader to load module somewhere else
- NullPage
 - Reserve first memory page in process, defense in depth
- EAF
 - Filter shellcode access to Export Address Table (kernel32 and ntdll)
- BottomUp Randomization
 - Randomize data structure bases

MORE EXPLOIT MITIGATIONS

- SEHOP-validate SEH chain looking for `_EXCEPTION_REGISTRATION` structure whose prev pointer is -1
- ROP Hardening (new in 4.0)
 - Deep Hooks-protect critical APIs and the APIs they call
 - AntiDetours-protect against jumping over detoured part of a function
 - Banned Functions-disallow calling `ntdll!LdrHotpatchRoutine`

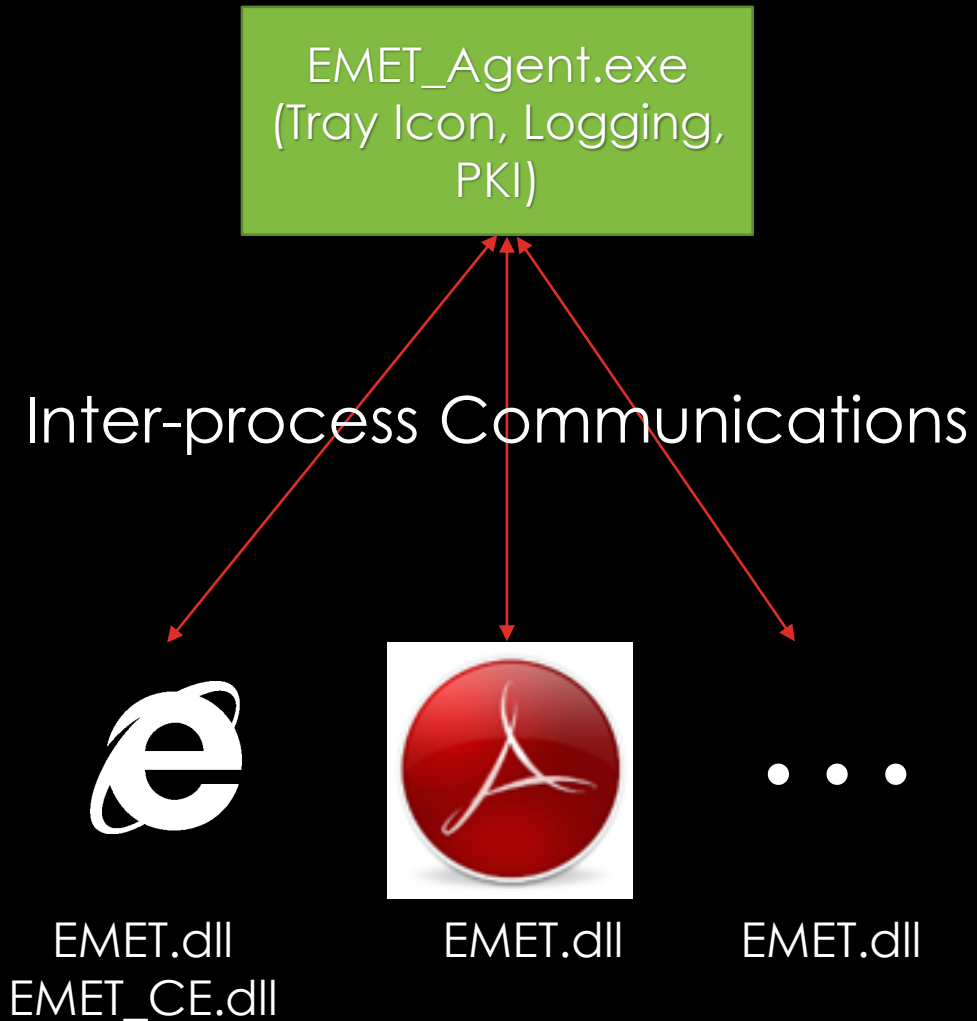
ROP

- Stands for **R**eturn **O**riented **P**rogramming
- Bypasses DEP (Data Execution Prevention)
- Attacker call stack injected into user controlled portion of memory
- Attacker stack has return pointers that point to “gadgets” in executable modules loaded in memory
- These specifically selected gadgets have a few instructions followed by a `ret`
 - These `ret` instructions then return to the location pointed to by the next pointer in the attacker stack
- Functions like `VirtualProtect` are commonly ROP'ed to
- Requires “Stack Pivot” to make x86 ESP register point to attacker call stack

ROP MITIGATIONS (NEW IN 4.0)

- ROP (Detour functions that are commonly ROP'ed to)
 - LoadLib
 - Make sure we are not trying to call LoadLibrary() on a network location
 - MemProt
 - Make sure we aren't making stack pages executable
 - Caller
 - Make sure return address on stack was proceeded by a call
 - Make sure we didn't ret to this function
 - SimExecFlow
 - Make sure we don't ret to ROP gadgets
 - StackPivot
 - Make sure Stack Pointer (ESP) is between stack limits defined by TIB

EMET ARCHITECTURE



WHAT IS PKI?

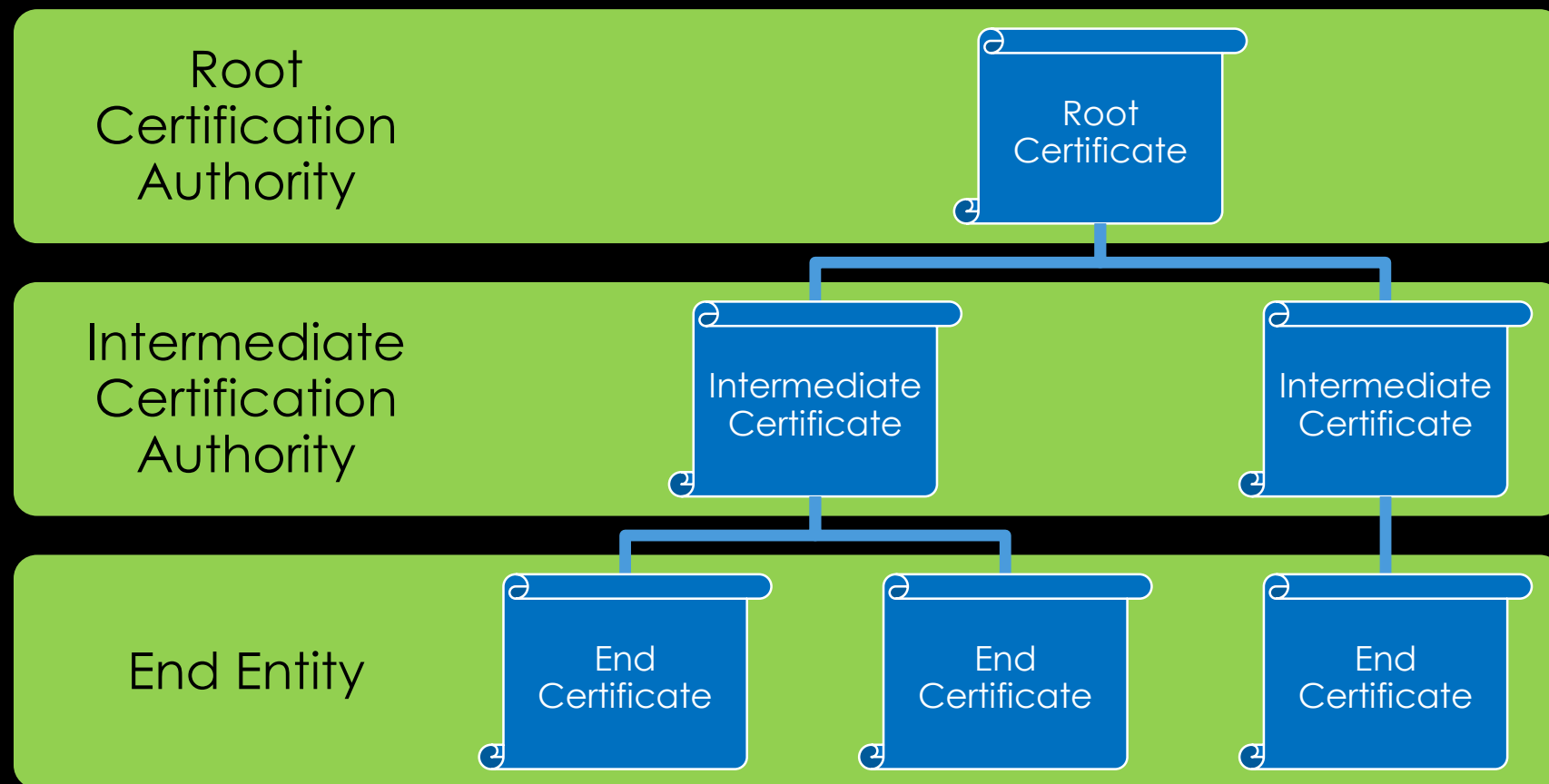
- A **public-key infrastructure (PKI)** is a set of hardware, software, people, policies, and procedures needed to create, manage, distribute, use, store, and revoke digital certificates.

--Wikipedia

- Used to ensure confidentiality, integrity and attribution online
- Communication with bank websites and other secure communications online depend on PKI
- PKI is the basis of HTTPS

PKI HIERARCHY

We can verify that each certificate is signed by a parent by looking for a digital signature of the parent on the child certificate.



RECENT SSL/TLS INCIDENTS

- December 2008- MD5 proven weak (Sotirov/Stevens)
- March 2011- Comodo CA signs 9 fraudulent certificates
- August 2011- Diginotar signs at least 1 fraudulent certificate
- November 2011- DigiCert issues 22 certs with 512 bit keys
- January 2013- TURKTRUST creates 2 issues fraudulent CAs and a certificate

PKI is under ATTACK

PKI CERTIFICATE PINNING

Pinning is when we enforce certain assumptions or expectations about certificates that we get from the internet



RELATED WORK

- TACK (Marlinspike, Perrin): requires TLS changes, pins to TACK signing key
- Convergence (Marlinspike): based on Perspectives project, new protocol
- DANE/TLS (RFC 6698): requires DNS changes
- HSTS (RFC 6797) + Draft ietf websec key pinning (Evans, Palmer, Sleevi): pins to SubjectPublicKeyInfo hash, requires HTTP changes, used in Chrome

EMET'S DESIGN GOALS

- Our goals in EMET PKI design were the following:
 1. Give control to users
 - Users specify the certificates
 - Users specify the domain names
 - Users specify the heuristic checks
 2. Don't require changes to pre-existing protocols/new protocols
 - This could break something
 - This would require adoption by the rest of the internet
 3. Keep EMET as a standalone tool on the client and not depend on remote services
- In order to achieve these goals, we had to make tradeoffs that existing designs didn't have to make

EMET'S APPROACH

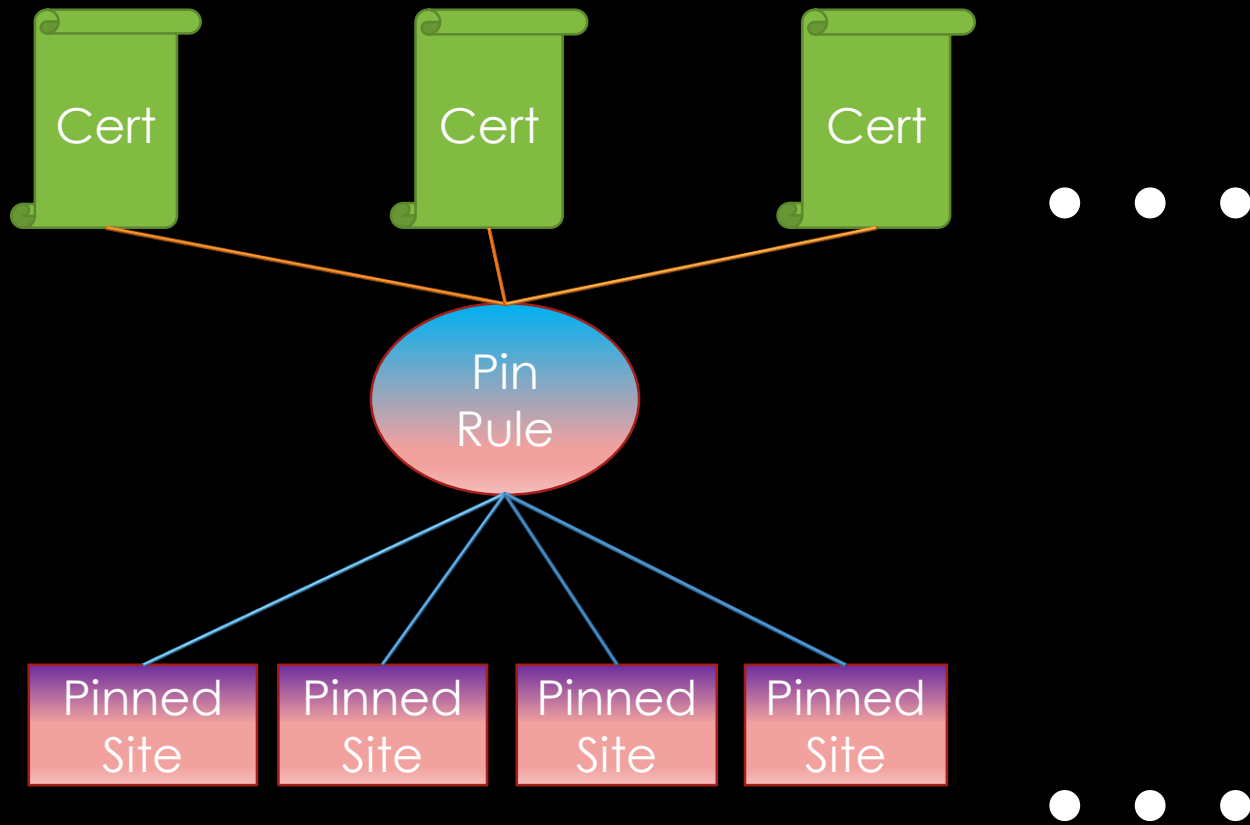
- Requires no protocol changes
 - Pins to Root Certificates, not Intermediate Certificates
 - Pins to certificates in the Current User's "Trusted Root Certification Authorities" store
 - Identifies certificates by either:
 - <Issuer, Serial #> Tuple
- OR**
- Subject Key Identifier (SHA-1 of subjectPublicKey)

CERTIFICATE IDENTIFICATION

- Certificates can be identified by <issuer, serial #> tuples
 - According to RFC5280:
 - “the issuer name and serial number identify a unique certificate”
 - Identifying a specific certificate is more rigid (restrictive)
- Certificates can be identified by Public Key
 - Some certificates chain to roots which share the same public key
 - EMET optionally allows certificate identification by only Subject Key Identifier (SHA-1 of hash of Public Key)

EMET PKI PINNING ARCHITECTURE

Architecture



Default Configuration Example

Baltimore CyberTrust Root
Verisign
GlobalSign
GTE CyberTrust Global Root

MSSkypeCA

login.skype.com
secure.skype.com

WINDOWS CAPI EXTENSION

- Implemented in EMET_CE[64].dll
- EMET_CE.dll loaded inside the process
- Communicates with EMET_Agent.exe, and passes it the entire certificate chain including the Root and End certificates hex encoded in XML
- EMET_Agent.exe decides whether the cert is OK or not

```
CryptRegisterOIDFunction() is called with following parameters:  
CRYPT_OID_VERIFY_CERTIFICATE_CHAIN_POLICY_FUNC,  
CERT_CHAIN_POLICY_SSL,  
EXPORT_FUNC_NAME
```

CERTIFICATE CHECKS 1

- If none of the following matches a Pinned Site's Domain Name, pass because this domain is not configured
 - Server Name of HTTPS connection
 - End certificate's Subject Name
 - End certificate's Subject Simple Name
 - End certificate's Subject DNS Name
 - End certificate's Subject URL Name
 - Any Subject Alternative Name on End certificate
- Is Pin Rule Expired?
 - If yes, fail

CERTIFICATE CHECKS 2

- Either (Depending on Configuration)
 - Is Subject Name of root **AND** Serial Number of root equal to that in a pinned Root Store certificate?
 - If yes, pass

OR

- Is root Subject Key Identifier equal to that in a pinned Root Store certificate?
 - If yes, pass

CERTIFICATE CHECKS 3 (EXCEPTIONS)

- Is root Public Modulus Bit length < Pin Rule's allowed length?
 - If yes, fail
- Is root Digest Algorithm disallowed by the Pin Rule?
 - If yes, fail
- Is root country equal to the Pin Rule's Allowed Country?
 - If no, fail


DEFAULT PROTECTED DOMAINS

- Shipped in CertTrust.xml
- Enabled by “Recommended Settings” in wizard
- Microsoft Protected Domains:
 - login.microsoftonline.com
 - secure.skype.com
 - login.live.com
 - login.skype.com
- 3rd Party Protected Domains:
 - www.facebook.com
 - login.yahoo.com
 - twitter.com

LIMITATIONS

- Mitigation is specifically for SSL/TLS
- Since we just check End and Root Certificates, we don't run heuristics on intermediate certificates
- Pin configuration is statically shipped with EMET, so they could get outdated
- If spoofed certificate chained to same root certificate as original, it might not be caught

- EMET's mitigations are not 100% "bullet proof"
 - They just try to raise the bar for attackers



DEMO TIME!

Recorded or Live?

REFERENCES

- ntdll!LdrHotpatchRoutine
 - <http://cansecwest.com/slides/2013/DEP-ASLR%20bypass%20without%20ROP-JIT.pdf>
- MD5 Harmful (Sotirov/Stevens)
 - <http://www.win.tue.nl/hashclash/rogue-ca/>
- TACK (Marlinspike, Perrin)
 - <http://tack.io/draft.html>
- Convergence (Marlinspike)
 - <http://convergence.io/>
- DANE/TLS RFC 6698
 - <http://tools.ietf.org/html/rfc6698>
- HSTS RFC 6797
 - <http://tools.ietf.org/html/rfc6797>
- Chrome's Public Key Pinning Extension (Evans, Palmer, Sleevi)
 - <http://tools.ietf.org/html/draft-ietf-websec-key-pinning-07>
- X509 RFC 5280
 - <http://tools.ietf.org/html/rfc5280>
- More Information about Memory Corruption Mitigations in EMET 4.0:
 - <http://www.recon.cx/2013/slides/Recon2013-Elias%20Bachalany-Inside%20EMET%204.pdf>
- ROP Explanation:
 - <http://www.neilscomputerblog.blogspot.com/2013/04/rop-return-oriented-programming.html>
 - <http://www.youtube.com/watch?v=Vyi8b3VOw9M>

Download EMET 4.0:
<http://www.microsoft.com/en-us/download/details.aspx?id=39273>

QUESTIONS ABOUT EMET

?