Welcome to

3. SDLC and Risk Ranking

KEA Kompetence OB2 Software Security 2019

Henrik Lund Kramshøj hlk@zencurity.com @kramse

Slides are available as PDF, kramse@Github
3-SDLC-and-risk-ranking.tex in the repo security-courses

Plan for today

Subjects

- Software Development Lifecycle
- Secure Software Development Lifecycle
- Phases of SSDL
- Roles and Responsibilities

Exercises

- Choose a few real vulnerabilities, prioritize them

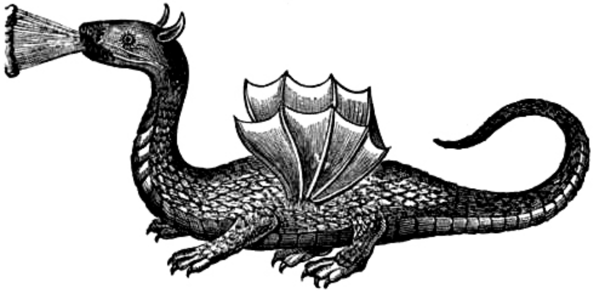
Reading Summary

AoST chapters 3: The Secure Software Development Lifecycle

AoST chapters 4: Risk-based Security Testing

AoST chapters 5: Shades of Analysis: white, Gray, and Black Box Testing

Goals:



Here be dragons

- Software is insecure
- How do we improve quality
- Higher quality is more stable, and more secure
- Make sure to test specifically for security issues

We talked about security design with Qmail and Postfix recently. This year has been bad for Exim mailserver: CVE-2019-10149, CVE-2019-13917 and CVE-2019-15846

Exim RCE CVE-2019-10149 June

VULNERABILITY PATCHED... BY ACCIDENT ...

This was only recently discovered by the Qualys team while auditing older Exim versions. Now, Qualys researchers are warning Exim users to update to the 4.92 version to avoid having their servers taken over by attackers. Per the same June 2019 report on email server market share, only 4.34% of all Exim servers run the latest 4.92 release.

In an email to Linux distro maintainers, Qualys said the vulnerability is ”trivially exploitable”and expects attackers to come up with exploit code in the coming days.

This Exim flaw is currently tracked under the CVE-2019-10149 identifier, but Qualys refers to it under the name of ”Return of the WIZard”because the vulnerability resembles the ancient WIZ and DEBUG vulnerabilities that impacted the Sendmail email server back in the 90s.

`https://www.zdnet.com/article/new-rce-vulnerability-impacts-nearly-half-of-the-internets-email-servers/`

See also detailed information from the finders:
`https://www.qualys.com/2019/06/05/cve-2019-10149/return-wizard-rce-exim.txt`

Exim RCE CVE-2019-10149 July

Issue: A local or remote attacker can execute programs with root privileges - if you've an unusual configuration. For details see below.

`https://exim.org/static/doc/security/CVE-2019-13917.txt`

Not enabled in default config!

Exim RCE CVE-2019-15846 September

The Exim mail transfer agent (MTA) software is impacted by a critical severity vulnerability present in versions 4.80 up to and including 4.92.1.

The bug allows local or unauthenticated remote attackers to execute programs with root privileges on servers that accept TLS connections.

The flaw tracked as CVE-2019-15846 — initially reported by ’Zerons’ on July 21 and analyzed by Qualys’ research team — is ”exploitable by sending an SNI ending in a backslash-null sequence during the initial TLS handshake”which leads to RCE with root privileges on the mail server.

https://www.bleepingcomputer.com/news/security/critical-exim-tls-flaw-lets-attackers-remotel

https://git.exim.org/exim.git/blob_plain/2600301ba6dbac5c9d640c87007a07ee6dcea1f4:/doc/doc-t
cve-2019-15846/cve.txt

Software Development Lifecycle

> A full lifecycle approach is the only way to achieve secure software.
> –Chris Wysopal

- Often security testing is an afterthought
- Vulnerabilities emerge during design and implementation
- Before, during and after approach is needed

Secure Software Development Lifecycle

- SSDL represents a structured approach toward implementing and performing secure software development
- Security issues evaluated and addressed early
- During business analysis
- through requirements phase
- during design and implementation

Functional specification needs to evaluate security

- Completeness
- Consistency
- Feasibility
- Testability
- Priority
- Regulations

Source: The Art of Software Security Testing Identifying Software Security Flaws Chris Wysopal ISBN: 9780321304865

Phases of SSDL

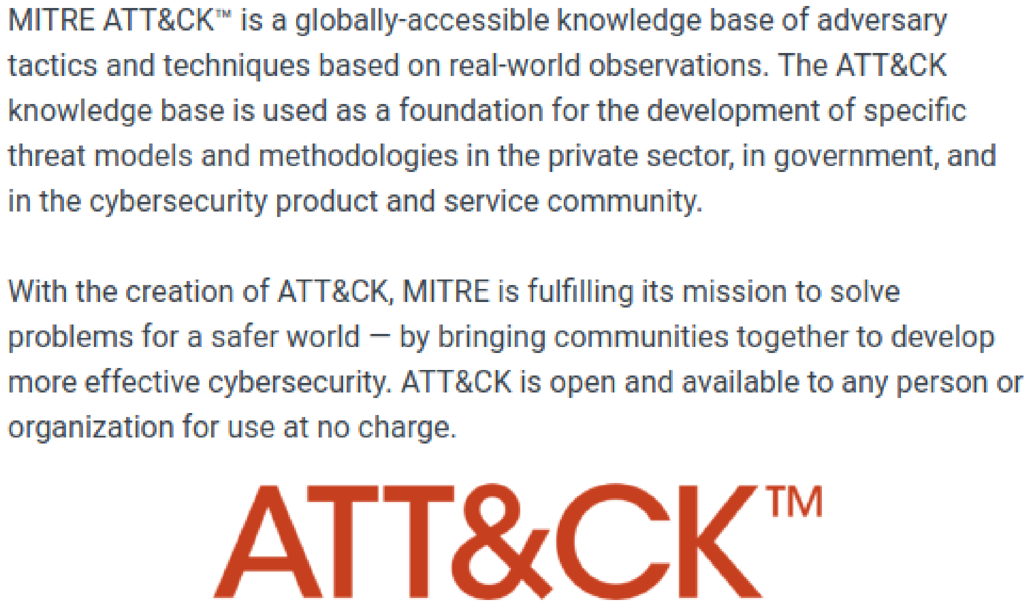
- Phase 1: Security Guidelines, Rules, and Regulations
- Phase 2: Security requirements: attack use cases
- Phase 3: Architectural and design reviews/threat modelling
- Phase 4: Secure coding guidelines
- Phase 5: Black/gray/white box testing
- Phase 6: Determining exploitability

Secure deployment comes next after this.

Phase 1: Security Guidelines, Rules, and Regulations

- *Umbrella requirement*
- Government regulations Sarbanes-Oxley Act (SOX)
- Payment regulations Payment Card Industry (PCI)
- OWASP, HIPAA, FISMA, BASEL II, ...
- ISO/IEC 27001 - information security management system standards http://en.wikipedia.org/wiki/ISO/IEC_27001
- SSAE 16 No. 16, Reporting on Controls at a Service Organization Statement on Standards for Attestation Engagements (SSAE) http://ssae16.com/
- ISAE 3402 Assurance Reports on Controls at a Service Organization International Standard on Assurance Engagements (ISAE) http://isae3402.com/

Phase 2: Security requirements: attack use cases



MITRE ATT&CK™ is a globally-accessible knowledge base of adversary tactics and techniques based on real-world observations. The ATT&CK knowledge base is used as a foundation for the development of specific threat models and methodologies in the private sector, in government, and in the cybersecurity product and service community.

With the creation of ATT&CK, MITRE is fulfilling its mission to solve problems for a safer world — by bringing communities together to develop more effective cybersecurity. ATT&CK is open and available to any person or organization for use at no charge.

ATT&CK™

- Does application store personal and/or sensitive information, health, HIPAA, GDPR
- MITRE ATT&CK framework may help `https://attack.mitre.org/`

Phase 3: Architectural and design reviews/threat modelling

- Help avoid insecure architectures and low-security design
- Threat modelling - a whole subject in itself
- Identify security critical parts of the application

Phase 4: Secure coding guidelines

- Plan use of static and dynamic analysis tools
- Train for secure coding
- Lay down rules for coding, dont use `strcpy` only `strlcpy`

Secure Coding Best Practices Handbook from Veracode

- **#01 Verify for Security Early and Often**
- #02 Parameterize Queries
- #03 Encode Data
- #04 Validate All Inputs
- #05 Implement Identity and Authentication Controls
- #06 Implement Access Controls
- #07 Protect Data
- #08 Implement Logging and Intrusion Detection
- #09 Leverage Security Frameworks and Libraries
- #10 Monitor Error and Exception Handling

https://info.veracode.com/secure-coding-best-practices-hand-book-guide-resource.html

Phase 5: Black/gray/white box testing

- Plan for testing
- Allow time for testing - critical part
- Continous integration may help to avoid pitfalls like, we are out of time – we will skip security testing

Phase 6: Determining exploitability

Ideally every vulnerability would be fixed

Determining exploitability is a factor in estimating risk associated

- Access needed to attempt exploitation
- Level of access or privilege yielded by successful exploitation
- The time or work factor required to exploit the vulnerability
- The exploits potential reliability
- The repeatability of exploit attempts

Deploying Applications Securely

- Having secure defaults helps
- Good initial file permissions
- Make sure application can be patched
- Track and prioritize identified vulnerabilities
- Make it easy to report vulnerabilities to the organization

Roles and Responsibilities

- Make it clear who has responsibility for security at various phases
- Program or product manager should write the security policies
- Product or project manager also responsible for certification processes
- Architects and developers are responsible for providing design and implementation
- QA/testers drive critical analyses of the system and build tests
- Security process managers oversee threat modelling, security assessments, and secure coding training

- Not an exhaustive list!

Risk-Based Security Testing

> Focus testing on areas where difficulty of attack is least and the impact is highest.
> –Chris Wysopal

Time and resources are constrained

Software development must be prioritized

Threat modelling / risk modelling exist to help this

- Identify threat paths
- Identify threats
- Identify vulnerabilities
- Rank/prioritize the vulnerabilities

Sounds easy enough, harder to do

DREAD

Book mentions DREAD model

> READ is part of a system for risk-assessing computer security threats previously used at Microsoft and although currently used by OpenStack and other corporations[citation needed] it was abandoned by its creators [1]. It provides a mnemonic for risk rating security threats using five categories.

The categories are:

- Damage – how bad would an attack be?
- Reproducibility – how easy is it to reproduce the attack?
- Exploitability – how much work is it to launch the attack?
- Affected users – how many people will be impacted?
- Discoverability – how easy is it to discover the threat?

Source: `https://en.wikipedia.org/wiki/DREAD_(risk_assessment_model)`

but was abandoned by Microsoft

Microsoft Secure Development Lifecycle

There are five major threat modeling steps:

- Defining security requirements.
- Creating an application diagram.
- Identifying threats.
- Mitigating threats.
- Validating that threats have been mitigated. Threat modeling should be part of your routine development lifecycle, enabling you to progressively refine your threat model and further reduce risk.

Sources:
`https://www.microsoft.com/en-us/securityengineering/sdl`
`https://www.microsoft.com/en-us/securityengineering/sdl/threatmodeling`

Example applications from Microsoft

Microsoft has released sample applications.

> Secure Development Documentation Learn how to develop and deploy secure applications on Azure with our sample apps, best practices, and guidance.
> Get started Develop a secure web application on Azure

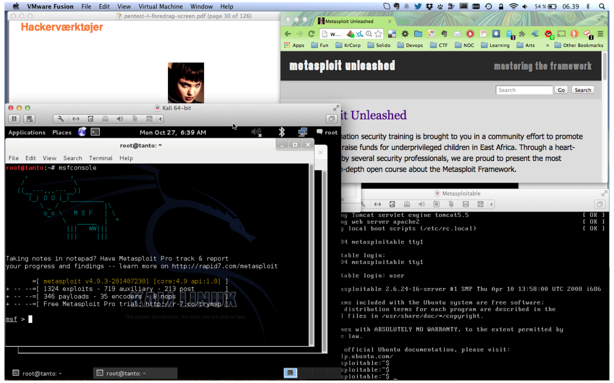
Source: `https://docs.microsoft.com/en-us/azure/security/develop/`

Yes, this describes how to run Alpine Linux on their Azure Cloud.

Blackbox, greybox og whitebox

- Forudsætninger og forudgående kendskab til miljøet
- Black Box testen involverer en sikkerhedstestning af et netværk uden nogen form for insider viden om systemet udover den IP-adresse, der ønskes testet. Dette svarer til den situation en fjendtlig hacker vil stå i og giver derfor det mest realistiske billede af netværkets sårbarhed overfor angreb udefra. Men er dårlig ressourceudnyttelse.
- I den anden ende af skalaen har vi White Box testen. I dette tilfælde har sikkerhedsspecialisten både før og under testen fuld adgang til alle informationer om det scannede netværk. Analysen vil derfor kunne afsløre sårbarheder, der ikke umiddelbart er synlige for en almindelig angriber. En White Box test er typisk mere omfattende end en Black Box test og forudsætter en højere grad af deltagelse fra kundens side, men giver en meget detaljeret og tilbundsgående undersøgelse.
- En Grey Box test er som navnet siger et kompromis mellem en White Box og en Black Box test. Typisk vil sikkerhedsspecialisten udover en IP-adresse være i besiddelse af de mest grundlæggende systemoplysninger: Hvilken type af server der er tale om (mail-, webserver eller andet), operativsystemet og eventuelt om der er opstillet en firewall foran serveren.

Testing Labs



it Unleashed

ation security training is brought to you in a community effort to promote raise funds for underprivileged children in East Africa. Through a heart- by several security professionals, we are proud to present the most -depth open course about the Metasploit Framework.

```
root@tanto:~# msfconsole

Taking notes in notepad? Have Metasploit Pro track & report
your progress and findings -- learn more on http://rapid7.com/metasploit

       =[ metasploit v4.9.3-2014072301 [core:4.9 api:1.0] ]
+ -- --=[ 1324 exploits - 719 auxiliary - 213 post       ]
+ -- --=[ 346 payloads - 35 encoders - 8 nops            ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf > 
```

```
ng Tomcat servlet engine tomcat5.5                              [ OK ]
ng web server apache2                                           [ OK ]
g local boot scripts (/etc/rc.local)                            [ OK ]

04 metasploitable tty1

table login:
04 metasploitable tty1

table login: user

asploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

ams included with the Ubuntu system are free software;
 distribution terms for each program are described in the
l files in /usr/share/doc/*/copyright.

mes with ABSOLUTELY NO WARRANTY, to the extent permitted by
e law.

 official Ubuntu documentation, please visit:
lp.ubuntu.com/
sploitable:~$
sploitable:~$
sploitable:~$ _
```

- Sniffers Wireshark and similar tools
- Proxies and fuzzers
- Debuggers
- Virtualisation - can also emulate ARM on Intel etc.
- Laptops and network hardware - dont use a HUB! Cheap managed switch with mirror port is better

Exercise

Now lets do the exercise

Real Vulnerabilities

which is number **11** in the exercise PDF.

For Next Time

Think about the subjects from this time, write down questions

Check the plan for chapters to read in the books
Most days have less than 100 pages, but some days may have more!

Visit web sites and download papers if needed

Retry the exercises to get more confident using the tools

Buy the books!