Advanced Web Attacks and Exploitation (OSWE)

Prof. Joas Antonio

Sobre o Autor

- Cyber Security Analyst, Cyber and Information Security Consultant by Betta GP, Information Security Researcher by Experience Security, Ethical Hacking and PenTest, OWASP Member and Researcher, Cybrary Teacher Assistant, Microsoft Instructor, Web Developer, Bug Hunter by HackerOne and OBB, Python Developer, has over +300 technology courses and +31 certifications, SANS Member, CIS Member and Research, Cyber Security Mentor and IT lover.

Sobre o Livro

- Fundamentos
- Iniciantes a Especialistas
- Livro 100% Hands-on
- Livro totalmente feito de referências, então a prática você vai ter através de blogs, tutoriais e vídeos
- Livro com fins didáticos
- Objetivo do livro é dar uma base de exploração web, baseada no curso OSWE da Offensive Security
- Necessário conhecimento em inglês (Traduzir conteúdo só atrasaria o seu desenvolvimento profissional)

Conceitos básicos sobre Web

- Para você ter maior compreensão, eu recomendo ler esses 2 ebooks feitos por mim
- https://drive.google.com/drive/folders/12Mvq6kE2HJDwN2CZhEGWizyWt87YunkU (Ataques web básico)
- E a apostila de ataques web do curso do Boot, mas não tem nenhum material copiado dele, somente a base
- https://drive.google.com/file/d/1MDR24V5xe5B_u7MnePLLjbliEsgjQx1P/view?usp=sharing

OSWE – FUNDAMENTOS E PRÁTICA

Objetivo

- O curso tem três objetivos principais: analisar o código-fonte, fazer engenharia reversa para aplicativos fechados (descompilar e depurar) e melhorar o pensamento para obter uma visão ampliada dos vetores padrão.

Laboratórios

- Infelizmente os laboratórios é baseado em outra certificação a OSCE ao qual tenho livro CTP (Crack the perimeter), então como dito antes, você precisa depurar eles para achar as vulnerabilidades.
- Além disso, as ferramentas que você utiliza muito são:
  - Burp Suite
  - Jd-gui
  - Grep
  - dnSPY
  - cURL
  - Wget

  "Conheça muito bem essas ferramentas", além de conhecer sistemas Linux

Ementa

- Tools & Methodologies.
- Persistent Cross-Site Scripting
- Blind SQL Injection.
- Type Juggling.
- Authentication Bypass.
- Cross-Site Request Forgery.
- Data Exfiltration.
- Bypassing File Upload Restrictions.
- Bypassing REGEX restrictions.
- .NET Deserialization.
- Session Hijacking.
- Source Code Recovery.

Preparando seu ambiente

- Kali Linux: Baixe o https://www.kali.org/ (Não obrigatório)
- Backup dos seus arquivos de estudo: https://www.youtube.com/watch?v=BvLMQMjV9YE
- Baixe o Ubuntu Server: https://ubuntu.com/download/server
- Baixe o Metasploitable: https://sourceforge.net/projects/metasploitable/
- Laboratórios práticos: https://www.vulnhub.com/
- Laboratórios práticos 2: https://www.hackthebox.eu/

BurpSuite Fundamentals

- Vamos pegar os fundamentos da ferramenta Burp Suite
- BurpSuite Proxy: https://portswigger.net/burp/documentation/desktop/tools/proxy/using
- BurpSuite Scope: https://portswigger.net/burp/documentation/desktop/tools/target/scope
- BurpSuite Comparer: https://portswigger.net/burp/documentation/desktop/tools/comparer
- BurpSuite Decoder: https://portswigger.net/burp/documentation/desktop/tools/decoder
- Proxy: https://support.portswigger.net/customer/portal/articles/1783055-configuring-your-browser-to-work-with-burp
- Scope: https://www.youtube.com/watch?v=K_92lb0k9FU
- Comparer e Repeater: https://www.youtube.com/watch?v=YSDJnRakx_E / https://www.youtube.com/watch?v=_sG4w2XECh8
- Decode: https://www.youtube.com/watch?v=8FMdEGDShmw

Web Interact with Python

- https://docs.python.org/2/library/simplehttpserver.html
- https://docs.python.org/3/library/http.server.html
- https://cleitonbueno.com/python-webserver-em-um-minuto/
- https://learn.adafruit.com/raspipe-a-raspberry-pi-pipeline-viewer-part-2/miniature-web-applications-in-python-with-flask
- https://towardsdatascience.com/controlling-the-web-with-python-6fceb22c5f08

Gerenciamento de código

- O controle do **código**-fonte (ou controle de versões) é a prática de monitoramento e **gerenciamento** de alterações no **código**.
- https://docs.microsoft.com/pt-br/dotnet/standard/managed-code
- https://gaea.com.br/como-melhorar-o-gerenciamento-de-codigo-fonte/
- https://aws.amazon.com/pt/devops/source-control/
- https://www.devmedia.com.br/como-adotar-a-analise-estatica-de-codigo/32727
- https://blog.locaweb.com.br/desenvolvedores/conheca-3-ferramentas-e-sites-que-avaliam-a-qualidade-do-codigo/
- https://blog.onedaytesting.com.br/auditoria-de-codigo/
- Compilador Java: https://www.youtube.com/watch?v=__qTvntdZIA

XSS Persistente

- Ataques armazenados são aqueles em que o script injetado é permanentemente armazenado nos servidores de destino, como em um banco de dados, em um fórum de mensagens, log de visitantes, campo de comentários etc. A vítima recupera o script malicioso do servidor quando solicita o armazenamento. em formação. O XSS armazenado também é conhecido como XSS Persistente ou Tipo I.
- https://www.welivesecurity.com/br/2017/07/07/vulnerabilidade-cross-site-scripting/
- https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)
- Pratica 1: https://www.youtube.com/watch?v=9WxtpOUGV8U
- Webinar XSS: https://www.youtube.com/watch?v=jSAswDmJ1o0
- Pratica 2: https://www.youtube.com/watch?v=xKcgkYkoaRs

XSS to RCE (OSWE Pratico)

- Exploit: https://www.exploit-db.com/exploits/20009
- Details: https://vulmon.com/vulnerabilitydetails?qid=CVE-2012-2593
- https://www.inc0.net/forum/forum/varie/exploit/9950-exploit-db-remote-atmail-email-server-appliance-6-4-xss-csrf-rce
- https://s0md3v.github.io/xss-to-rce/
- https://blog.ripstech.com/2019/mybb-stored-xss-to-rce/
- https://medium.com/@knownsec404team/the-analysis-of-mybb-18-20-from-stored-xss-to-rce-7234d7cc0e72
- https://github.com/xapax/xss-to-rce

Session Hijacking

- Session hijacking é a exploração de uma sessão de computador válida, às vezes também chamada de uma chave de sessão - para obter acesso não autorizado a informações ou serviços em um sistema de computador.
- Prática 1: https://www.youtube.com/watch?v=D--gCvOS59g
- Prática 2: https://www.youtube.com/watch?v=P_u3g95bzlE
- https://www.youtube.com/watch?v=g5vyj85Gxd4
- https://www.owasp.org/index.php/Session_hijacking_attack
- https://www.youtube.com/watch?v=kh30ylrpU68

Session Riding (CSRF)

- A falsificação de solicitação entre sites (CSRF) é um ataque que força um usuário final a executar ações indesejadas em um aplicativo Web no qual eles estão atualmente autenticados. Os ataques CSRF visam especificamente solicitações de alteração de estado, não roubo de dados, pois o invasor não tem como ver a resposta à solicitação forjada. Com uma pequena ajuda da engenharia social (como o envio de um link por email ou bate-papo), um invasor pode induzir os usuários de um aplicativo da Web a executar ações de sua escolha. Se a vítima for um usuário normal, um ataque CSRF bem-sucedido pode forçar o usuário a executar solicitações de alteração de estado, como transferência de fundos, alteração de endereço de email e assim por diante. Se a vítima for uma conta administrativa, o CSRF poderá comprometer todo o aplicativo da web.

Session Riding (CSRF) PT 2

Mais informações

- https://www.paladion.net/blogs/session-riding-attacks
- https://crypto.stanford.edu/cs155old/cs155-spring08/papers/Session_Riding.pdf
- http://shiflett.org/blog/2005/session-riding
- https://security.stackexchange.com/questions/138650/comparing-session-hijacking-fixation-and-riding
- https://www.youtube.com/watch?v=KaEj_qZgiKY
- https://www.secpoint.com/cross-site-request-forgery.html

Session Riding (SSRF) PT 3

- A falsificação de solicitação do lado do servidor (também conhecida como SSRF) é uma vulnerabilidade de segurança da Web que permite que um invasor induza o aplicativo do servidor a fazer solicitações HTTP para um domínio arbitrário de sua escolha.
- Em exemplos típicos de SSRF, o invasor pode fazer com que o servidor faça uma conexão de volta para si mesmo ou para outros serviços baseados na Web na infraestrutura da organização ou para sistemas externos de terceiros.
- https://portswigger.net/web-security/ssrf
- https://www.youtube.com/watch?v=66ni2BTIjS8 Prática
- https://www.youtube.com/watch?v=_IVjvNelzMw Prática 2
- https://www.owasp.org/index.php/Server_Side_Request_Forgery

REMOTE CODE EXECUTION

- Uma vulnerabilidade de **execução arbitrária de código** é uma falha de segurança em software ou hardware que permite a execução arbitrária de código. Um programa projetado para explorar essa vulnerabilidade é chamado de exploração de **execução de código arbitrária** . A capacidade de acionar a execução arbitrária de código em uma rede (especialmente por meio de uma rede de área ampla, como a Internet) é geralmente chamada de RCE (**execução remota de código**).

REMOTE CODE EXECUTION (PRATICO)

- **Bypass de autenticação do ATutor e RCE (2.2.1) CVE-2016-2555**
- Instale: https://sourceforge.net/projects/atutor/files/atutor_2_2_1/
- https://www.exploit-db.com/exploits/39514
- https://srcincite.io/advisories/src-2016-0009/
- https://www.exploit-db.com/exploits/39639
- https://github.com/atutor/ATutor/commit/d74f1177cfa92ed8e49aa65f724f308b4a3ac5b9

Data Exfiltration

- A exfiltração de dados ocorre quando um malware e / ou um agente malicioso realiza uma transferência de dados não autorizada de um computador. Também é comumente chamado extrusão ou exportação de dados. A exfiltração de dados também é considerada uma forma de roubo de dados.
- https://fluxguard.com/how-to-guides/detect-data-exfiltration-from-xss-and-javascript-injection-attacks
- https://www.pentestpartners.com/security-blog/data-exfiltration-techniques/
- https://azeria-labs.com/data-exfiltration/
- https://martinojones.com/data-exfiltration-using-valid-icmp-packets-b5c489548fb1?gi=8d18695075e1
- https://www.patternex.com/threatex/detecting-and-verifying-icmp-exfiltration-with-ai-enabled-platform
- https://blogs.akamai.com/2017/09/introduction-to-dns-data-exfiltration.html
- https://sqlwiki.netspi.com/attackQueries/dataExfiltration/#mysql
- https://www.sqreen.com/plugins/mysql-data-exfiltration

ATutor LMS Type Juggling Vulnerability

- Subvertendo o ATutor Authentication:
- Instale: https://sourceforge.net/projects/atutor/files/atutor_2_2_1/
- https://srcincite.io/advisories/src-2016-0012/
- https://github.com/sourceincite/poc/blob/master/SRC-2016-0012.py
- https://github.com/atutor/ATutor/commit/2eed42a74454355eddc7fc119e67af40dba1a94c
- Reference: PHP Type Juggling
  - https://www.youtube.com/watch?v=ASYuK01H3Po
  - https://www.netsparker.com/blog/web-security/type-juggling-authentication-bypass-cms-made-simple/

Entendendo a desserialização de Java

- Para entender a desserialização (ou desserialização), precisamos entender a primeira serialização.
- Cada aplicativo lida com dados, como informações do usuário (por exemplo, nome de usuário, idade) e os utiliza para executar ações diferentes: executar consultas SQL, fazer logon em arquivos (tenha cuidado com o GDPR) ou apenas exibi-las. Muitas linguagens de programação oferecem a possibilidade de trabalhar com objetos para que os desenvolvedores possam agrupar dados e métodos em classes.
- Serialização é o processo de converter os dados do aplicativo (como objetos) em um formato binário que pode ser armazenado ou enviado pela rede, para ser reutilizado pelo mesmo ou por outro aplicativo, que o desserializará como um processo reverso.
- A idéia básica é que é fácil criar e reutilizar objetos.

https://nytrosecurity.com/2018/05/30/understanding-java-deserialization/

Exemplo prático: https://github.com/wetw0rk/AWAE-PREP/tree/master/Understanding%20Java%20Deserialization

JavaScript para PenTest

- Curso: https://www.pentesteracademy.com/course?id=11
- Exemplo: https://github.com/wetw0rk/AWAE-PREP/tree/master/JavaScript%20For%20Pentesters
- https://subscription.packtpub.com/book/networking_and_servers/9781783988525/6/ch06lvl1sec39/xss-and-javascript-a-deadly-combination
- https://blog.appsecco.com/static-analysis-of-client-side-javascript-for-pen-testers-and-bug-bounty-hunters-f1cb1a5d5288

SQL Injection to RCE

- Laboratório: https://pentesterlab.com/exercises/from_sqli_to_shell/course
- https://medium.com/bugbountywriteup/sql-injection-to-lfi-to-rce-536bed29a862
- https://blog.ripstech.com/2019/dotcms515-sqli-to-rce/
- https://pwnrules.com/flickr-from-sql-injection-to-rce/
- https://www.youtube.com/watch?v=JgoN3sDad04
- https://www.youtube.com/watch?v=JZR8bDRl0t8

PHP Object Injection

- O PHP Object Injection é uma vulnerabilidade no nível do aplicativo que pode permitir que um invasor execute diferentes tipos de ataques maliciosos, como Injeção de Código , Injeção de SQL , Path Traversal e Negação de Serviço de Aplicativo , dependendo do contexto. A vulnerabilidade ocorre quando a entrada fornecida pelo usuário não é higienizada adequadamente antes de ser passada para a função PHP unserialize (). Como o PHP permite a serialização de objetos, os invasores podem transmitir seqüências serializadas ad-hoc para uma chamada unserialize () vulnerável, resultando em uma injeção arbitrária de objetos PHP no escopo do aplicativo.
- https://www.owasp.org/index.php/PHP_Object_Injection
- https://securitycafe.ro/2015/01/05/understanding-php-object-injection/
- https://nitesculucian.github.io/2018/10/05/php-object-injection-cheat-sheet/
- Exemplo: https://github.com/wetw0rk/AWAE-PREP/tree/master/Understanding%20PHP%20Object%20Injection

OWASP PROJECT

- https://www.owasp.org/index.php/OWASP_Broken_Web_Applications_Project
- O OWASP Broken Web Applications Project é um conjunto de aplicativos Web vulneráveis distribuídos em uma Máquina Virtual.
- O projeto Broken Web Applications (BWA) produz uma máquina virtual executando uma variedade de aplicativos com vulnerabilidades conhecidas para aqueles interessados em:
- aprendendo sobre segurança de aplicativos da web
- teste de técnicas de avaliação manual
- teste de ferramentas automatizadas
- teste de ferramentas de análise de código fonte
- observando ataques na web
- testando WAFs e tecnologias de código similares
- Enquanto isso, poupamos as pessoas interessadas em aprender ou testar a dor de ter que compilar, configurar e catalogar todas as coisas normalmente envolvidas na execução desse processo do zero.

Bypass File Upload Restriction

- Os arquivos enviados representam um risco significativo para os aplicativos. O primeiro passo em muitos ataques é obter algum código no sistema a ser atacado. Então o ataque precisa apenas encontrar uma maneira de executar o código. O uso de um upload de arquivo ajuda o invasor a executar a primeira etapa.
- As consequências do upload irrestrito de arquivos podem variar, incluindo controle completo do sistema, um sistema de arquivos ou banco de dados sobrecarregado, encaminhamento de ataques para sistemas de back-end, ataques do lado do cliente ou desfiguração simples. Depende do que o aplicativo faz com o arquivo carregado e, principalmente, de onde está armazenado.
- Existem realmente duas classes de problemas aqui. O primeiro é com os metadados do arquivo, como o caminho e o nome do arquivo. Eles geralmente são fornecidos pelo transporte, como codificação HTTP com várias partes. Esses dados podem induzir o aplicativo a substituir um arquivo crítico ou a armazená-lo em um local incorreto. Você deve validar os metadados com muito cuidado antes de usá-los.
- A outra classe de problemas está no tamanho ou no conteúdo do arquivo. A variedade de problemas aqui depende inteiramente do uso do arquivo. Veja os exemplos abaixo para obter algumas idéias sobre como os arquivos podem ser mal utilizados. Para se proteger contra esse tipo de ataque, você deve analisar tudo o que seu aplicativo faz com arquivos e pensar cuidadosamente sobre o processamento e os intérpretes envolvidos.
- https://www.exploit-db.com/docs/english/45074-file-upload-restrictions-bypass.pdf
- https://pentestlab.blog/2012/11/29/bypassing-file-upload-restrictions/

Bypass File Upload Restriction

- Os arquivos enviados representam um risco significativo para os aplicativos. O primeiro passo em muitos ataques é obter algum código no sistema a ser atacado. Então o ataque precisa apenas encontrar uma maneira de executar o código. O uso de um upload de arquivo ajuda o invasor a executar a primeira etapa.
- As consequências do upload irrestrito de arquivos podem variar, incluindo controle completo do sistema, um sistema de arquivos ou banco de dados sobrecarregado, encaminhamento de ataques para sistemas de back-end, ataques do lado do cliente ou desfiguração simples. Depende do que o aplicativo faz com o arquivo carregado e, principalmente, de onde está armazenado.
- Existem realmente duas classes de problemas aqui. O primeiro é com os metadados do arquivo, como o caminho e o nome do arquivo. Eles geralmente são fornecidos pelo transporte, como codificação HTTP com várias partes. Esses dados podem induzir o aplicativo a substituir um arquivo crítico ou a armazená-lo em um local incorreto. Você deve validar os metadados com muito cuidado antes de usá-los.
- A outra classe de problemas está no tamanho ou no conteúdo do arquivo. A variedade de problemas aqui depende inteiramente do uso do arquivo. Veja os exemplos abaixo para obter algumas idéias sobre como os arquivos podem ser mal utilizados. Para se proteger contra esse tipo de ataque, você deve analisar tudo o que seu aplicativo faz com arquivos e pensar cuidadosamente sobre o processamento e os intérpretes envolvidos.
- https://www.exploit-db.com/docs/english/45074-file-upload-restrictions-bypass.pdf
- https://pentestlab.blog/2012/11/29/bypassing-file-upload-restrictions/

Bypass File Upload Restriction

- https://null-byte.wonderhowto.com/how-to/bypass-file-upload-restrictions-using-burp-suite-0164148/
- https://sushant747.gitbooks.io/total-oscp-guide/bypass_image_upload.html
- Pratica 1: https://www.youtube.com/watch?v=Ue3wtxR9s0E
- Pratica 2:https://www.youtube.com/watch?v=SDRJHbnmjhw

ManageEngine Applications Manager AMUserResourcesSyncServlet SQL Injection RCE

- Instale: http://archives.manageengine.com/applications_manager/12900
- https://manageenginesales.co.uk/2018/05/manageengine-applications-manager-build-13730-released/
- https://www.postgresql.org/docs/9.4/functions-binarystring.html
- https://www.mulesoft.com/tcat/tomcat-jsp
- Extra: Deserialization Vulnerability
  - https://www.geeksforgeeks.org/serialization-in-java/
  - https://github.com/frohoff/ysoserial
  - https://blog.jamesotten.com/post/applications-manager-rce/

Codificação Dupla

- Essa técnica de ataque consiste em codificar os parâmetros de solicitação do usuário duas vezes no formato hexadecimal para ignorar os controles de segurança ou causar comportamento inesperado no aplicativo. É possível porque o servidor da web aceita e processa solicitações de clientes de várias formas codificadas.
- Ao usar a codificação dupla, é possível ignorar os filtros de segurança que decodificam a entrada do usuário apenas uma vez. O segundo processo de decodificação é executado pela plataforma ou módulos de back-end que manipulam corretamente os dados codificados, mas não possuem as verificações de segurança correspondentes.
- Os invasores podem injetar codificação dupla em nomes de caminho ou cadeias de consulta para ignorar o esquema de autenticação e os filtros de segurança em uso pelo aplicativo Web.
- Existem alguns conjuntos de caracteres comuns usados nos ataques de aplicativos da Web. Por exemplo, os ataques do Path Traversal usam "../" (barra de pontos e pontos), enquanto os ataques XSS usam caracteres "<" e ">". Esses caracteres fornecem uma representação hexadecimal que difere dos dados normais.
- Por exemplo, os caracteres "../" (barra com ponto) representam% 2E% 2E% 2f na representação hexadecimal. Quando o símbolo% é codificado novamente, sua representação no código hexadecimal é% 25. O resultado do processo de codificação dupla "../"(dot-dot-slash) seria% 252E% 252E% 252F:
- A codificação hexadecimal de "../" representa "% 2E% 2E% 2f"
- A codificação de "%" representa "% 25"
- A codificação dupla de "../" representa "% 252E% 252E% 252F"

Postgresql Extension

- https://www.cybertec-postgresql.com/en/secure-postgresql-a-reminder-on-various-attack-surfaces/
- https://www.cvedetails.com/vulnerability-list/vendor_id-336/product_id-575/Postgresql-Postgresql.html
- https://github.com/dhamaniasad/awesome-postgres
- https://wiki.postgresql.org/wiki/A_Guide_to_CVE-2018-1058%3A_Protect_Your_Search_Path
- https://www.youtube.com/watch?v=WIBPq4jeZaA

UDF Reverse Shell

- Durante um teste de penetração, podemos nos jogar em uma situação em que temos apenas acesso administrativo ao SQL. Como sempre, queremos mergulhar mais fundo na rede. Às vezes, a única maneira de conseguir isso é executar comandos no sistema que atende o servidor SQL atual.
- Se o servidor for um MSSQL, a maneira mais simples de fazer isso é aproveitar o procedimento armazenado xp_cmdshell. O pior cenário seria se o xp_cmdshell estivesse desabilitado, mas isso pode ser desfeito facilmente com esta consulta: sp_configure 'xp_cmdshell', '1'
- O tópico desta publicação, no entanto, é adicionar mais uma arma importante ao nosso arsenal no caso de um servidor MySQL, onde não há xp_cmdshell ou procedimento armazenado equivalente; nós estaremos falando sobre a UDF (Função Definida pelo Usuário) no MySQL. Mais especificamente, criaremos um UDF que executa comandos do sistema através de um servidor MySQL.
- “Nos bancos de dados SQL, uma função definida pelo usuário fornece um mecanismo para estender a funcionalidade do servidor de banco de dados, adicionando uma função que pode ser avaliada nas instruções SQL.” - http://en.wikipedia.org/wiki/User-defined_function
- Para que o mecanismo UDF funcione, as funções que serão usadas devem ser escritas em C ou C ++.

UDF Reverse Shell

- https://www.obrela.com/blog/article/using-udf-penetration-testing/
- https://securitypentester.ninja/mysql-udf-injection/
- https://osandamalith.com/2018/02/11/mysql-udf-exploitation/
- https://github.com/rapid7/metasploit-framework/blob/master/documentation/modules/exploit/multi/mysql/mysql_udf_payload.md
- https://infamoussyn.wordpress.com/2014/07/11/gaining-a-root-shell-using-mysql-user-defined-functions-and-setuid-binaries/

Bassmaster NodeJS Arbitrary JavaScript Injection Vulnerability (1.5.1) CVE-2014-7205

- Instale: npm install bassmaster@1.5.1
- https://www.npmjs.com/package/bassmaster
- https://www.rapid7.com/db/modules/exploit/multi/http/bassmaster_js_injection
- https://github.com/rapid7/metasploit-framework/blob/master/modules/exploits/multi/http/bassmaster_js_injection.rb
- https://www.exploit-db.com/exploits/40689
- https://vulners.com/nodejs/NODEJS:337
- https://medium.com/@Bank_Security/undetectable-c-c-reverse-shells-fab4c0ec4f15
- https://ired.team/offensive-security-experiments/offensive-security-cheetsheets
- https://www.metahackers.pro/reverse-shells-101/

DotNetNuke Cookie Deserialization RCE (<9.1.1) CVE-2017-9822

- Instale: https://github.com/dnnsoftware/Dnn.Platform/releases/tag/v9.1.0
- https://www.blackhat.com/docs/us-17/thursday/us-17-Munoz-Friday-The-13th-Json-Attacks.pdf
- https://media.blackhat.com/bh-us-12/Briefings/Forshaw/BH_US_12_Forshaw_Are_You_My_Type_WP.pdf
- https://gist.github.com/pwntester/72f76441901c91b25ee7922df5a8a9e4
- https://paper.seebug.org/365/
- https://www.youtube.com/watch?v=oUAeWhW5b8c
- https://vulners.com/seebug/SSV:96326
- https://www.slideshare.net/MSbluehat/dangerous-contents-securing-net-deserialization
- https://www.exploit-db.com/docs/english/44756-deserialization-vulnerability.pdf

XmlSerializer Limitations

- https://gist.github.com/SamuelEnglard/978742401960aae3eaa7e95a27c0d63b
- https://www.codeproject.com/Articles/15646/A-Deep-XmlSerializer-Supporting-Complex-Classes-En

XmlSerializer Limitations

- https://dotnetnukeru.com/dnndocs/api/html/M_DotNetNuke_Common_Utilities_FileSystemUtils_PullFile.htm
- https://dotnetnukeru.com/dnn6docs/api/html/T_DotNetNuke_Common_Utilities_FileSystemUtils.htm
- https://www.blackhat.com/docs/us-17/thursday/us-17-Munoz-Friday-The-13th-Json-Attacks.pdf

Desserialização

- https://cheatsheetseries.owasp.org/cheatsheets/Deserialization_Cheat_Sheet.html
- https://community.microfocus.com/t5/Security-Research-Blog/New-NET-deserialization-gadget-for-compact-payload-When-size/ba-p/1763282
- https://social.msdn.microsoft.com/Forums/vstudio/en-US/3268fd25-4a1d-46af-82ad-edcdb555de69/limitations-of-xmlserializer-what-objects-cannot-be-serialized?forum=csharpgeneral

YSOSerial.NET

- O ysoserial.net é uma coleção de "cadeias de gadgets" de programação orientada a propriedades e utilitários descobertas em bibliotecas .NET comuns que podem, nas condições certas, explorar aplicativos .NET executando desserialização insegura de objetos. O programa principal do driver pega um comando especificado pelo usuário e o agrupa na cadeia de gadgets especificada pelo usuário e, em seguida, serializa esses objetos no stdout. Quando um aplicativo com os gadgets necessários no caminho de classe desserializa esses dados sem segurança, a cadeia será automaticamente invocada e fará com que o comando seja executado no host do aplicativo.
- Deve-se notar que a vulnerabilidade está no aplicativo executando desserialização insegura e NÃO em ter gadgets no caminho de classe.
- https://github.com/pwntester/ysoserial.net
- https://pt.slideshare.net/cisoplatform7/automated-discovery-of-deserialization-gadget-chains-117547762

REGEX

- Uma expressão regular é uma notação para representar padrões em strings. Serve para validar entradas de dados ou fazer busca e extração de informações em textos.
- Por exemplo, para verificar se um dado fornecido é um número de 0,00 a 9,99 pode-se usar a expressão regular \d,\d\d, pois o símbolo \d é um curinga que casa com um dígito.
- http://turing.com.br/material/regex/introducao.html
- https://regexr.com/
- https://pt.wikipedia.org/wiki/Express%C3%A3o_regular
- https://medium.com/trainingcenter/entendendo-de-uma-vez-por-todas-express%C3%B5es-regulares-parte-7-66be1ac1f72d
- https://regex101.com/

LFI – Local File Inclusion

- A falha de **local file inclusion** permite que o atacante inclua um arquivo para explorar o mecanismo de dynamic file inclusion(inclusão dinâmica de arquivo) implementado na aplicação web. A falha ocorre devido ao fato de que o atacante pode passar qualquer valor para o parâmetro da aplicação alvo e a mesma não faz a validação correta do valor informado antes de executar a operação. Esse tipo de falha faz com que a aplicação web mostre o conteúdo de alguns arquivos, mas dependendo da severidade, essa falha também permite:
- – Execução de código no servidor
  – Execução de código no client-side. Por exemplo, JavaScript, o que pode levar a ocorrência de outros tipos de ataques como XSS por exemplo
  – Negação de Serviço(DoS)
  – Vazamento de informações sensíveis
- https://www.infosec.com.br/local-file-inclusion-remore-file-inclusion/
- https://www.owasp.org/index.php/Testing_for_Local_File_Inclusion
- https://www.youtube.com/watch?v=kcojXEwolls

XXE – XML ATTACK

- Um ataque de *entidade externa XML* é um tipo de ataque contra um aplicativo que analisa a entrada XML. Esse ataque ocorre quando a **entrada XML que contém uma referência a uma entidade externa é processada por um analisador XML mal configurado** . Esse ataque pode levar à divulgação de dados confidenciais, negação de serviço, falsificação de solicitação do servidor, varredura de portas na perspectiva da máquina em que o analisador está localizado e outros impactos no sistema.
- https://www.owasp.org/index.php/XML_External_Entity_(XXE)_Processing
- http://labs.siteblindado.com/2019/02/xml-external-entity-xxe.html
- https://github.com/payloadbox/xxe-injection-payload-list
- https://medium.com/@klose7/xxe-attacks-part-2-xml-dtd-related-attacks-a572e8deb478

Testing for HTTP Verb Tampering

- A especificação HTTP inclui métodos de solicitação diferentes dos pedidos GET e POST padrão. Um servidor Web compatível com os padrões pode responder a esses métodos alternativos de maneiras não previstas pelos desenvolvedores. Embora a descrição comum seja adulteração de 'verbo', o padrão HTTP 1.1 refere-se a esses tipos de solicitação como métodos 'HTTP' diferentes.
- https://www.owasp.org/index.php/Testing_for_HTTP_Verb_Tampering_(OTG-INPVAL-003)
- https://www.acunetix.com/vulnerabilities/web/http-verb-tampering/
- https://www.imperva.com/learn/application-security/http-verb-tampering/

Man in the Browser

- O ataque Man-in-the-Browser é a mesma abordagem que o ataque Man-in-the-middle , mas, neste caso, um Trojan Horse é usado para interceptar e manipular chamadas entre o executável do aplicativo principal (por exemplo, o navegador) e seus mecanismos de segurança ou bibliotecas on-the-fly.
- O objetivo mais comum desse ataque é causar fraude financeira, manipulando transações de sistemas de Internet Banking, mesmo quando outros fatores de autenticação estão em uso.
- Um cavalo de Tróia instalado anteriormente é usado para agir entre o navegador e o mecanismo de segurança do navegador, detectando ou modificando transações à medida que são formadas no navegador, mas ainda exibindo a transação pretendida pelo usuário.
- Normalmente, a vítima deve ser inteligente para perceber o sinal de um ataque enquanto está acessando um aplicativo da Web como uma conta bancária na Internet, mesmo na presença de canais SSL, porque todos os controles e mecanismos de segurança esperados são exibidos e funcionam normalmente.
- Pontos de efeito:
- **Objetos auxiliares do navegador** - bibliotecas carregadas dinamicamente e carregadas pelo Internet Explorer na inicialização
- **Extensões** - o equivalente a objetos auxiliares do navegador do navegador Firefox
- **API Hooking** - esta é a técnica usada pelo Man-in-the-Browser para executar seu Man-in-the-Middle entre o aplicativo executável (EXE) e suas bibliotecas (DLL).
- **Javascript** - Usando um worm Ajax malicioso, conforme descrito no Ajax Sniffer - Prova de conceito

Man in the Browser

- https://www.owasp.org/index.php/Man-in-the-browser_attack
- https://en.wikipedia.org/wiki/Man-in-the-browser
- https://codesealer.com/the-secrets-behind-man-in-the-browser-attacks/
- https://www.youtube.com/watch?v=cm0wqPUv6mQ

LDAP Injection

- O LDAP (Lightweight Directory Access Protocol) é usado para armazenar informações sobre usuários, hosts e muitos outros objetos. A injeção LDAP é um ataque no servidor, que pode permitir que informações confidenciais sobre usuários e hosts representados em uma estrutura LDAP sejam divulgadas, modificadas ou inseridas. Isso é feito através da manipulação de parâmetros de entrada passados posteriormente para funções internas de pesquisa, adição e modificação.
- Um aplicativo da web pode usar LDAP para permitir que os usuários se autentiquem ou pesquisem as informações de outros usuários dentro de uma estrutura corporativa. O objetivo dos ataques de injeção LDAP é injetar metacaracteres dos filtros de pesquisa LDAP em uma consulta que será executada pelo aplicativo.
- https://www.owasp.org/index.php/Testing_for_LDAP_Injection_(OTG-INPVAL-006)
- https://www.blackhat.com/presentations/bh-europe-08/Alonso-Parada/Whitepaper/bh-eu-08-alonso-parada-WP.pdf
- https://github.com/swisskyrepo/PayloadsAllTheThings/tree/master/LDAP%20Injection
- https://www.youtube.com/watch?v=wtahzm_R8e4
- https://www.youtube.com/watch?v=iUbqJy_MOiE

XPATH Injection

- Semelhante à injeção de SQL , os ataques de injeção de XPath ocorrem quando um site usa informações fornecidas pelo usuário para construir uma consulta XPath para dados XML. Ao enviar informações intencionalmente malformadas para o site, um invasor pode descobrir como os dados XML estão estruturados ou acessar dados aos quais ele normalmente não pode ter acesso. Ele pode até conseguir elevar seus privilégios no site se os dados XML estiverem sendo usados para autenticação (como um arquivo de usuário baseado em XML).
- A consulta ao XML é feita com o XPath, um tipo de instrução descritiva simples que permite que a consulta XML localize uma parte da informação. Como o SQL, você pode especificar certos atributos a serem encontrados e padrões a serem correspondidos. Ao usar XML para um site, é comum aceitar alguma forma de entrada na sequência de consultas para identificar o conteúdo a ser localizado e exibido na página. Essa entrada **deve** ser higienizada para verificar se não atrapalha a consulta XPath e retorna os dados incorretos.
- XPath é um idioma padrão; sua notação / sintaxe é sempre independente da implementação, o que significa que o ataque pode ser automatizado. Não há dialetos diferentes, pois ocorre em solicitações para os bancos de dados SQL.
- Como não há controle de acesso de nível, é possível obter o documento inteiro. Não encontraremos nenhuma limitação, como sabemos nos ataques de injeção de SQL.

XPATH Injection

- https://www.owasp.org/index.php/XPATH_Injection
- http://projects.webappsec.org/w/page/13247005/XPath%20Injection
- https://www.youtube.com/watch?v=fV0qsqcScI4
- https://www.youtube.com/watch?v=5ZDSPVp1TpM

SMTP Injection

- Essa ameaça afeta todos os aplicativos que se comunicam com servidores de correio (IMAP / SMTP), geralmente aplicativos de webmail. O objetivo deste teste é verificar a capacidade de injetar comandos arbitrários de IMAP / SMTP nos servidores de correio, devido ao fato de os dados de entrada não serem adequadamente higienizados.
- A técnica de injeção IMAP / SMTP é mais eficaz se o servidor de email não estiver diretamente acessível na Internet. Onde for possível a comunicação completa com o servidor de correio back-end, é recomendável realizar testes diretos.
- Uma injeção de IMAP / SMTP possibilita o acesso a um servidor de correio que, de outra forma, não seria acessível diretamente da Internet. Em alguns casos, esses sistemas internos não têm o mesmo nível de segurança e proteção de infraestrutura que é aplicado aos servidores Web front-end. Portanto, os resultados do servidor de correio podem estar mais vulneráveis a ataques dos usuários finais
- https://www.owasp.org/index.php/Testing_for_IMAP/SMTP_Injection_(OTG-INPVAL-011)
- https://www.acunetix.com/blog/articles/email-header-injection/
- https://www.youtube.com/watch?v=paAJqEcAmEU
- https://www.youtube.com/watch?v=3JvonYzpmV8

Lista de algumas vulnerabilidades com foco web

Arbitrary file access
Binary planting
Blind SQL Injection
Blind XPath Injection
Brute force attack
Buffer overflow attack
Cache Poisoning
Cash Overflow
Clickjacking
Command injection attacks
Comment Injection Attack
Content Security Policy
Content Spoofing
Credential stuffing
Cross Frame Scripting
Cross Site History Manipulation (XSHM)
Cross Site Tracing
Cross-Site Request Forgery (CSRF)
Cross Site Port Attack (XSPA)
Cross-Site Scripting (XSS)
Cross-User Defacement
Custom Special Character Injection
Denial of Service
Direct Dynamic Code Evaluation ('Eval Injection')
Execution After Redirect (EAR)
Exploitation of CORS
Forced browsing
Form action hijacking
Format string attack
Full Path Disclosure
Function Injection
Host Header injection
HTTP Response Splitting
HTTP verb tampering
HTML injection
LDAP injection
Log Injection
Man-in-the-browser attack
Man-in-the-middle attack
Mobile code: invoking untrusted mobile code
Mobile code: non-final public field
Mobile code: object hijack
One-Click Attack
Parameter Delimiter
Page takeover
Path Traversal
Reflected DOM Injection
Regular expression Denial of Service – ReDoS
Repudiation Attack
Resource Injection

Lista de algumas vulnerabilidades com foco web

- Server-Side Includes (SSI) Injection
  Session fixation
  Session hijacking attack
  Session Prediction
  Setting Manipulation
  Special Element Injection
  SMTP injection
  SQL Injection
  SSI injection
  Traffic flood
  Web Parameter Tampering
  XPATH Injection
  XSRF or SSRF

Conclusão

- Essa é a base para que você expanda o seu conhecimento em exploração web de maneira avançada, pensando fora da caixa e saindo daquele aquário que te prende com vulnerabilidades comuns e básicas
- Esse é um conteúdo que reunir para você ter base, o resto dependerá de você ;)