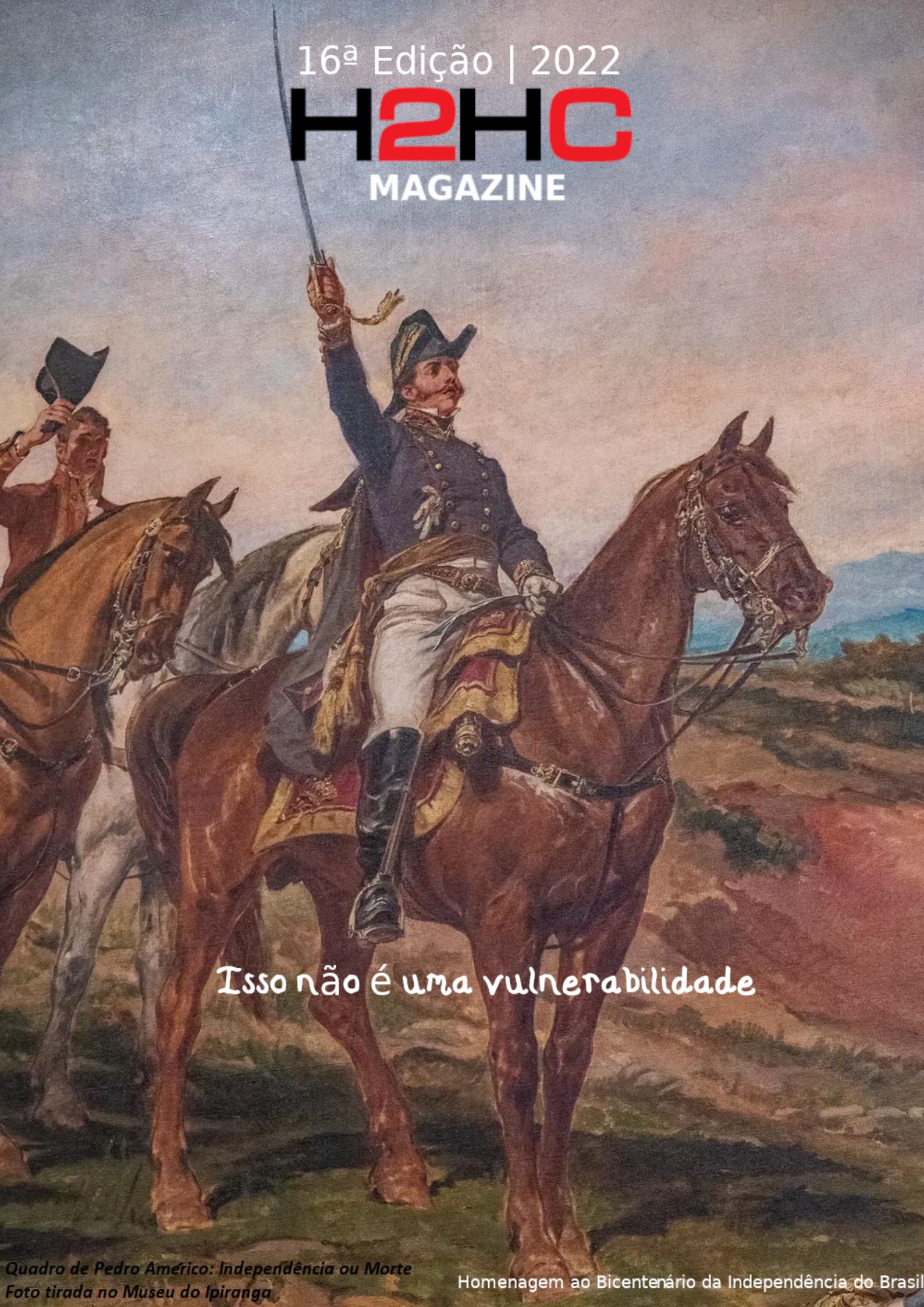


16^a Edição | 2022

H2HC MAGAZINE



Isso não é uma vulnerabilidade

H2HC

HACKERS TO HACKERS CONFERENCE

www.h2hc.com.br



CARTA DO EDITOR

Prezado(a) leitor(a),

É com grande satisfação que apresentamos a **16ª edição da H2HC Magazine!**

Além da versão online, essa revista também foi impressa e distribuída durante a H2HC de 2022.

Nessa edição não colocamos a coluna "Curiosidades" pois, por ser uma edição impressa, temos algumas restrições com relação ao número de páginas. Pelo mesmo motivo, também não adicionamos a coluna "O Exploit Que Eu Vi".

Gostaríamos de relembrar a todos que o principal objetivo dessa revista é contribuir com a comunidade de forma gratuita. Desta forma, estamos sempre a disposição para orientar pesquisas e escrita de artigos para qualquer pessoa, independente do nível de conhecimento e experiência - basta nos escrever!

A H2HC Magazine é totalmente comprometida com a qualidade das informações aqui publicadas. Se você encontrou algum erro ou gostaria de agregar alguma informação, por favor, entre em contato! Mensagens de apreciação ou crítica também são muito bem vindas e servem de estímulo ao nosso trabalho.

Nosso e-mail é revista@h2hc.com.br.

Boa leitura!

Editor

Gabriel Negreira Barbosa



@gabrielnb



SOBRE A H2HC MAGAZINE



H2HC MAGAZINE
16ª Edição | Outubro 2022

DIREÇÃO GERAL
Rodrigo Rubira Branco (BSDaemon)
Filipe Balestra

REDAÇÃO / REVISÃO TÉCNICA
Gabriel Negreira Barbosa
Rodrigo Rubira Branco (BSDaemon)

AGRADECIMENTOS
Fernando Mercês
Marcos Rosário (god6ixx)
Maycon Vitali (OutOfBound)
P3nDr1v3
Wendel Guglielmetti Henrique

Registro Único desta Edição (DOI)

<https://doi.org/10.47986/16>

Versão da Revista - Incrementada caso correções sejam lançadas

0.01

REDES SOCIAIS DO EVENTO



WEBSITE
<https://www.h2hc.com.br/revista>

CARTA DO EDITOR	3
SOBRE A H2HC MAGAZINE	4
Agenda	7
Palestras	9
A researcher's take on Spectre exploits	9
An Insight into Railway Security	9
Aprendizado de Maquina em Segurança	10
Breaking Firmware Trust from Pre EFI: Exploiting Early Boot Phases	10
Browser Exploitation and JIT Compilers	11
Cinema Time!	11
Data only Attacks Against UEFI BIOS	11
Desenvolvimento de um serviço de email privativo	12
Dont Blink! A deep dive into Cyclops Blink	12
Eleicoes Transparentes	13
Engenharia Reversa num jogo de Gameboy Advance	13
Exploitation tactics discovery using data analytics	13
From PIX to Reverse Shell	14
GitHub Actions: Vulnerabilities, Attacks and Counter Measures	14
Keynote by Alex Ionescu	15
Keynote: Ensaio sobre o Recrutamento	15
Retbleed: Arbitrary Speculative Code Execution with Return Instructions	15
State of the Art in IPv6 Attack & Defense	15
Techniques for publishing statistical information without violating individual's privacy according to the LGPD and GDPR	16
The ADCS domain administrator is right there	17
The Joy of Exploiting the Kernel on 2022	18
To branch or not to branch: security implications of x86 frontend implementations	18
Palestrantes	20
Alex Ionescu	20
Alexander Ermolov	20
Alexandra Sandulescu	20
Brian Butterly	21
Bruno Cortes	21
Bruno Macabeus	21
Edmond Rogers	22
Eduardo Vela	22
Evandro Hora	22
Fabricio Gimenes	22

Fernando Gont	23
Fernando Merces	23
Gustavo Scotti	23
Jeroen van de Graaf	23
Johannes Wikner	24
Jorge Buzeti	24
Lourenco A. Pereira	24
Magno Logan	25
Matrosov & Ermolov & Vasilenko & Thomas	25
Pawel Wieczorkiewicz	26
pirata & belphégor	26
Tarakanov & Labunets	26
Artigo NUCLEI - Scan rápido e versátil	27
Resumo	27
Introdução	27
Instalação e Execução	27
Templates	28
Cenário real	31
Conclusão	32
Referências	32
Artigo Web Cache Deception Attack	34
Introdução	34
Serviços de Cache em Web	34
Redes de Distribuição de Conteúdo (CDN)	34
Path Confusion	36
Web Cache Deception	37
Recomendações	38
Ataque Web Cache Deception in the Wild	39
Conclusão	40
Referências	40
Artigo Anti-x64dbg	41
Introdução	41
O bug	41
A técnica de detecção do x64dbg	43
Referências	45
Engenharia Reversa de Software	46
Reconstrução da IAT: Quando cortar thunks e excluir nodes?	46
Desabafo	46
Introdução	46
Cenário	47
Calculando distâncias	47
Investigando a IAT	49
Conclusão	51
Referências	51

Agenda

Dia 1

	H2HC	H2HC University
08:20–08:50	Credenciamento e entrega dos crachás	
08:50–09:10	Abertura - Filipe Balestra & Rodrigo Branco	
09:10–10:10	Keynote by Evandro Hora	
10:10–11:10	Data-only Attacks Against UEFI BIOS Alexander Ermolov	Aprendizado de Maquina em Segurança Lourenco A. Pereira
11:10–12:10	A researcher's take on Spectre exploits Alexandra Sandulescu	Engenharia Reversa num jogo de Gameboy Advance Bruno Macabeus
12:10–14:10	Lunch / Almoço	
14:10–15:10	Breaking Firmware Trust from Pre-EFI Matrosov & Ermolov & Vasilenko & Thomas	Dont Blink! A deep dive into Cyclops Blink Fernando Merces
15:10–16:10	State of the Art in IPV6 Attack & Defense Fernando Gont	Browser Exploitation and JIT Compilers Jorge Buzeti
16:10–16:40	Break / Intervalo	
16:40–17:40	An Insight into Railway Security Brian Butterly	Github Actions - Vulnerabilities, Attacks and Counter-measures Magno Logan



Dia 2

	H2HC	H2HC University
10:00-11:00		Keynote by Alex Ionescu
11:00-12:00	Retbleed - Arbitrary Speculative Code Execution with Return Instructions Johannes Wikner	The ADCS domain administrator is right there Fabricio Gimenes
12:00-14:00		Lunch / Almoço
14:00-15:00	Cinema Time! Tarakanov & Labunets	Desenvolvimento de um serviço de email privativo pirata & belphegor
15:00-16:00	To branch or not to branch - security implications of x86 frontend implementations Pawel Wieczorkiewicz	Techniques for publishing statistical information without violating individual's privacy according to the LGPD and GDPR Jeroen van de Graaf
16:00-16:30		Break / Intervalo
16:30-17:30	The Joy of Exploiting the Kernel in 2022 Eduardo Vela	From PIX to Reverse Shell Bruno Cortes
17:30-18:30	Exploitation tactics discovery using data analytics Edmond Rogers	Eleicoes Transparentes Gustavo Scotti
18:30-19:00		Encerramento



HACKERS TO HACKERS CONFERENCE

A researcher's take on Spectre exploits

Alexandra Sandulescu

Security Engineer, Google

Speculative execution attacks are (still) a hot topic because solutions are impractical, insufficient or both. Researching novel attack techniques, mitigation bypasses or new classes of attacks is a real roller coaster and might discourage people outside of academia. My talk discusses the building blocks of a Spectre exploit and how to make them more accessible for the broader security research public. The end goal is to make Spectre attacks practical and less complicated to pull off.

An Insight into Railway Security

Brian Butterly

Security Engineer

While being obvious for Security professionals, everybody is slowly but surely understanding that securing the IT worlds isn't sufficient. Thus, most companies are also applying their measure to other domains, like Operational Technology. One potentially even more specific area is the railway domain. From a Hacker's perspective trains are big, loud, cool, and fun. Sadly, rail is a very closed world, with specific tech that we only rarely get to touch.

During the presentation I will lift some of the fog surrounding the area and give various insights into where rail is really special and where things simply are just the way we as Hackers would expect.

The talk will give an overview of the following topics:

- Parts & Components of the overall railway system
- Current developments and directions
- Insights into regulatory requirements
 - The German approach, which should at least give some inspiration
- Processes and lifecycles
- Implications of being "special"

All in all the talk will give a bunch of inspiration for interested Hackers and researchers but also explain why caution is highly recommended.

Aprendizado de Maquina em Segurança

Lourenco A. Pereira

Professor, Instituto Tecnológico de Aeronáutica(ITA)

Segurança Cibernética é um elemento essencial para a transformação digital em que vivemos. Percebemos que cada vez mais a complexidade e a integração de sistemas computacionais é alta. Com isso, temos o desafio de entender profundamente as tecnologias para saber lidar com a descoberta e mitigação de ameaças e ações maliciosas que subvertem nossos sistemas. A situação de fragilidade em que estamos fica evidente quando observamos infraestruturas físicas em que sistemas computacionais provocam consequências cinéticas, causando baixas e danos nas mais variadas esferas de valor. Muito embora, o avanço tecnológico tenha proporcionado melhora na comunicação entre as pessoas e também promovido melhores serviços em geral; percebe-se que ainda há muita fragilidade nos sistemas e nos recursos humanos que desenvolvem estes sistemas.

Nesse contexto, é feita uma explanação sobre infraestruturas críticas e sistemas computacionais de modo que podemos estudá-las exaustivamente para caracterizar ataques conhecidos e detectar anomalias na carga de trabalho típica do sistema em análise. Assim,

Aprendizado de Máquina é apresentado como ferramental capaz de auxiliar neste processo pois possibilita uma avaliação em uma escala com ordens de magnitude maior daquela que um ser humano é capaz de lidar.

Breaking Firmware Trust from Pre EFI: Exploiting Early Boot Phases

Matrosov & Ermolov & Vasilenko & Thomas

Binarly Inc.

Vulnerabilities in System Management Mode (SMM) and more general UEFI applications/drivers (DXE) are receiving increased attention from security researchers. Over the last 9 months, the Binarly efiXplorer team disclosed 42 high-impact vulnerabilities related to SMM and DXE firmware components. But newer platforms have significantly increased the runtime mitigations in the UEFI firmware execution environment (including SMM). The new Intel platform firmware runtime mitigations reshaped the attack surface for SMM/DXE with new Intel Hardware Shield technologies applied below-the-OS.

The complexity of the modern platform security features is growing every year. The general security promises of the platform consist of many different layers defining their own security boundaries. Unfortunately, in many cases, these layers may introduce inconsistencies in mitigation technologies and create room for breaking general security promises, allowing for successful attacks.

In this presentation, we will share our work exploring recent changes in the UEFI firmware security runtime using one of the most recent Intel CPUs as an example. The presentation will cover the evolution of firmware mitigations in SMM/DXE on x86-based CPUs and a discussion about the new attacks on Intel Platform Properties Assessment Module (PPAM), which are often used in tandem with Intel SMI Transfer Monitor (STM).

These topics have never been publicly discussed from the offensive security research perspective.

Browser Exploitation and JIT Compilers

Jorge Buzeti

Cyber Security Analyst, ISH

Browsers são softwares extremamente complexos com dezenas de módulos e submódulos com responsabilidades vastas, desde renderização, execução de código, compiladores, JIT e entre outros. Por seu uso vasto e uma superfície de ataque mais vasta ainda, não é de se assustar uma intensa área de pesquisa em cada segmento de funcionamento dos navegadores.

Nessa palestra será abordado uma introdução a browser exploitation abordando o funcionamento geral de um navegador, a execução de JavaScript, o processo de compilação JIT, os principais tipos de bugs e técnicas e uma visão detalhada da CVE-2021-21220.

Cinema Time!

Tarakanov & Labunets

Independent Security Researchers

Media parsing is known as one of the weakest components of every consumer system. It often operates complex data structures in the most performant way possible, which is at odds with security requirements, such as attack surface minimization, compartmentalization, and privilege separation. Compared to other operating systems, video decoding on MacOS/iOS is an interesting case for two different reasons. First, instead of running in usermode, a considerable portion of format parsing is implemented in a kernel extension called AppleAVD, exposing the kernel to additional remote attack vectors. Second, recent anonymous reports suggest that AppleAVD may have been exploited in the wild. Our talk investigates AppleAVD kernel extension in-depth, covering video decoding subsystem internals, analysis of vulnerabilities, and ways to exploit them.

Data only Attacks Against UEFI BIOS

Alexander Ermolov

Principal Security Researcher, Binarly Inc.

What comes to your mind when you hear about UEFI BIOS vulnerabilities? For a long time the obvious answer was issues in SMM (System Management Mode) code, which enables one of the protection mechanisms against UEFI BIOS modifications. This was the reason of creation other platform protective technologies, but still new issues in SMM keep being discovered.

Though, supported not by each OEM/IBV, there are a number of mitigations applied for SMM code. Beyond that, a lot of firmware verification techniques were introduced recently. All measures grown by vendors aimed to protect the firmware code integrity and runtime UEFI BIOS interfaces (like SMI handlers) from

software attacks and hardware tampering. However, UEFI firmware architecture still allows to develop attack vectors that has almost none countermeasures nowadays and allows to bypass all known UEFI BIOS mitigations and protection technologies.

In this talk we'll describe current UEFI BIOS security model and talk about one of its main disadvantages, which could be exploited by recently discovered vulnerabilities.

Desenvolvimento de um serviço de email privativo

pirata & belphegor

Darkweb, Worldwide

Essa palestra visa discutir com a comunidade os principais detalhes técnicos (implementação segura de um servidor SMTP minimalista, análise de código, fortificação do ambiente, escolha de tecnologias como linguagens de programação e hardware, etc) e não técnicos (casos de uso, requisitos de segurança e privacidade, etc) por trás da criação de um serviço de e-mail temporário 100% gratuito e de código livre, com foco em privacidade e segurança. Esse é um projeto voluntário que visa prover para a comunidade uma alternativa segura e privativa para obter contas de e-mail, sem a necessidade de se registrar, para os mais diversos fins como cadastro em fóruns, serviços online, sites, etc.

Por mais que a palestra seja específica para o serviço de e-mail em questão, as ideias discutidas podem ser aplicadas a outros projetos de segurança nas mais variadas áreas da computação.

Dont Blink! A deep dive into Cyclops Blink

Fernando Merces

Senior Threat Researcher, Trend Micro

In 2022 Cyclops Blink became known by the world as the next attack from the well-known advanced persistent group Sandworm. Associated to destructive malware like BlackEnergy and Olympic Destroyer, this group also compromises IoT devices around the world to use it as their infrastructure. In 2018, VPNFilter was one such malware family that affected many routers globally from many different vendors – and consisted of multiple payloads and functions. After the industry sinkholed their domains, many infections were left over that could have been utilized by this group.

However, they chose instead to retool and attack new routers with malware that has been dubbed “Cyclops Blink”. In February 2022 NCSC in the UK published about WatchGuard specific Cyclops Blink attacks, and through our investigation Trend Micro was able to acquire different families of Cyclops Blink samples - one specifically attacking ASUS routers. Analyzing these samples, we were able to emulate an infection and track down and monitor more than 150 C&C servers from the threat actor infrastructure. While businesses around the world are spending time and money to stop attacks, nation state attackers are going after consumer devices to gain footholds for future attacks. How can we expect our parents to defend from being part of the next large scale nation attack if businesses already struggle?

Eleicoes Transparentes

Gustavo Scotti

Secure Firmware Engineer, Microsoft Corporation

Existira' um pais onde o processo das eleicoes publicam todos os logs de todas as urnas para que qualquer pessoa possa apurar os resultados. Partindo desse olhar, quais os requisitos tecnologicos e de segurança vao garantir que nenhum log seja alterado? Quais os problemas fundamentais das eleicoes? Que problemas existem hoje que possam ser resolvidos com tecnologia existente (block-chain), e quais os problemas que nao tem respostas? O foco da apresentacao e' o debate tecnico. Em nenhum momento irei comparar este cenario hipotetico com as eleicoes do Brasil, nem tampouco irei avaliar se a urna eletronica e' segura ou nao.

Engenharia Reversa num jogo de Gameboy Advance

Bruno Macabeus

Sr. Full-Stack Engineer, Anima

O Gameboy Advance foi um dos videogames mais populares de seu tempo, e com isso, muitas comunidades surgiram para estudar e documentar sua arquitetura, desenvolver ROM hacking, assim como ferramentas próprias para o GBA.

Então, que tal explorarmos engenharia reversa na prática com o seguinte desafio: desenvolver um editor de fases para um jogo de GBA, o "Klonoa: Empire of Dreams"? Esse é um desafio bem interessante, pois precisaremos entender a arquitetura de um hardware em ARM, aplicar engenharia reversa para descobrir a lógica do jogo, escrever patches para a ROM, e enfim usar todas as nossas descobertas para construir um completo editor de fases.

Veremos o passo a passo da engenharia reversa e o desenvolvimento da ferramenta nessa talk.

Repositório do projeto: <https://github.com/macabeus/klo-gba.js>

Manual sobre o desenvolvimento do projeto: <https://medium.com/@bruno.macabeus/pt-br-engenharia-reversa-num-jogo-de-gameboy-advance-introdu%C3%A7%C3%A3o-21ebffe2f794>

Exploitation tactics discovery using data analytics

Edmond Rogers

Academic Researcher

We have taken memory analysis tools and mapped file access interactions in kernel space. This mapping has allowed us to use a custom written implementation of GNN to visualize these memory interactions to provide baseline geometries to profile “weird machines” and post exploit tactics. In this talk we will introduce this research topic, provide code examples of this new GNN implementation, and discuss initial

findings.

From PIX to Reverse Shell

Bruno Cortes

Lead Offensive Security Engineer, Nubank

This is a hands-on presentation that addresses details about the BRCODE pattern adopted by BACEN for PIX transactions and demonstrates, based on a pentest finding scenario, the exploitation of Server Side Request Forgery (SSRF) + CRLF Injection -> Remote Code Execution (RCE) chained vulnerabilities. The flaw still affects financial institutions that don't apply the security measures established for PIX transactions with dynamic BRCODE.

GitHub Actions: Vulnerabilities, Attacks and Counter Measures

Magno Logan

Information Security Specialist, Trend Micro

This talk plans to demonstrate how GitHub Actions work and show security measures to protect your Actions from misuse by attackers. First, we'll do a deep dive into the Runners, the servers provided by GitHub to run your Actions, and the risks of using them. Then, we'll show how attackers can leverage these runners to mine cryptocurrencies, pivot into other targets, and more. Lastly, we'll demonstrate how to maliciously distribute backdoors into different repositories via the GitHub Actions Marketplace.

This presentation results from detailed research published earlier this year on the topic where the author investigated abuse case scenarios such as how attackers were leveraging this free service to mine cryptocurrencies on their behalf and behalf of other users, among other attack vectors. We'll also demonstrate how to perform interactive commands to the Runner servers via reverse shell, which are technically not allowed via traditional means. Ultimately, we'll show the problem of third-party dependencies via the GitHub Actions Marketplace. Showing how easy it is to create a fake GitHub Action that, if used unwillingly by other projects, can make their runners act as bots to target other victims and even be used in supply-chain attacks by tampering with the result of the pipeline.

Full research article:

<https://research.trendmicro.com/GitHubActions>

Follow-up article:

https://www.trendmicro.com/en_us/research/22/g/unpacking-cloud-based-cryptocurrency-miners-that-abuse-github-ac.html

Research repositories:

<https://github.com/magnologan/gha-test>

<https://github.com/magnologan/fake-gha>

Keynote by Alex Ionescu

Alex Ionescu

Technical Director of Platform Operations, CSE (Communications Security Establishment)

Keynote

Keynote: Ensaio sobre o Recrutamento

Evandro Hora

Socio-Fundador e Diretor da Tempest Security Intelligence

Reflexoes sobre os desafios em recrutar e reter talentos na area de Seguranca da Informacao

Retbleed: Arbitrary Speculative Code Execution with Return Instructions

Johannes Wikner

PhD student, ETH Zurich

Retbleed is the new addition to the family of speculative execution attacks that exploit branch target injection to leak arbitrary information on Intel and AMD CPUs. Unlike its siblings, who trigger harmful branch target speculation by exploiting indirect jumps or calls, Retbleed exploits return instructions with the same outcome. This means a great deal, since it undermines some of our current defenses.

State of the Art in IPv6 Attack & Defense

Fernando Gont

Security Consultant & Researcher, SI6 Networks

Many content providers (such as Google) report that over 40% of their network traffic in countries such as Brazil, the USA, or Germany is IPv6-based. Yet, IPv6 security implications are ignored or misunderstood by the vast majority of security professionals, leading to lousy IPv6 pentests and deficient IPv6 defenses.

Over the last few years, a number of advances have been made in IPv6 attack and defense, ranging from improved IPv6 network reconnaissance techniques and tooling, to privacy improvements in IPv6 addressing, resulting in IPv6 security becoming "a moving target".

In this presentation, Fernando Gont will provide a snapshot of the state of the art in IPv6 attack and defense, discussing the latest advancements in each of these areas, and providing concrete practical advice for both red teams and blue teams.

Techniques for publishing statistical information without violating individual's privacy according to the LGPD and GDPR

Jeroen van de Graaf

Professor, Universidade Federal de Minas Gerais

Cada vez mais são coletados dados contendo informações detalhadas sobre indivíduos: das suas pesquisas na internet, telefonemas, localização, saúde, genoma, etc. E cada vez mais esses dados estão sendo usados para análises estatísticas. No entanto, para evitar violações de privacidade, esses dados devem ser protegidos de uma forma ou outra, seja concedendo apenas acesso privilegiado à base de dados, seja modificando ou excluindo dados. Essas técnicas são conhecidas como controle de divulgação estatística.

Vários casos proeminentes mostraram que as técnicas mais óbvias, como a anonimização ou a pseudonimização, quase nunca funcionam. Isso foi confirmado num estudo da UFMG sobre a divulgação anual de dados estudantis do MEC/INEP. Além disso, muitas das técnicas mais sofisticadas (anonimato-k, diversidade-l, proximidade-t) também apresentam várias desvantagens. Isso se deve em parte ao fato de que definir a privacidade corretamente é de facto bastante difícil.

Apresentaremos a intuição por trás da *privacidade diferencial*, uma definição que incorpora quaisquer ataques futuros e, portanto, praticamente a única abordagem bem-sucedida na proteção da privacidade nesse contexto. Em essência, a privacidade diferencial consiste em adicionar sutilmente ruído aos dados, de modo que a privacidade individual seja protegida enquanto a inferência estatística significativa ainda é possível. Um problema aqui é que nem sempre se sabe de antemão qual fato estatístico será considerado interessante.

Claramente é impossível ter privacidade perfeita ("não publique nada") e inferência estatística perfeita ("publique tudo") ao mesmo tempo, então a privacidade diferencial oferece um equilíbrio calibrado entre os dois extremos. De fato, adicionar ruído a dados perfeitos e prístinos pode parecer estúpido, mas é a única maneira de preservar um mínimo de privacidade em um mundo orientado a dados. Desistir da privacidade não é uma opção, portanto escolhas terão que ser feitas. A sociedade precisa se conscientizar desse dilema e discutir os prós e os contras das várias soluções.

ABSTRACT

Increasingly data is collected containing detailed information about individuals: about internet searches, phone calls, location, health, genome, etc. And increasingly such data is being used for statistical analysis. However, to avoid privacy violations, this data should be protected one way or another, either by conceding only privileged access to the data base, or by modifying or deleting data. These techniques are known as statistical disclosure control.

Several high profile cases have shown that the most obvious techniques, like anonymization or pseudonymization, almost never work. This has been confirmed in UFMG's study of MEC/INEP annual release of student data. In addition, many of the more sophisticated techniques (k-anonymity, l-diversity, t-proximity) have several drawbacks too. This is in part due to the fact that defining privacy correctly is actually quite difficult.

We will present the intuition behind *differential privacy*, a definition which incorporates any future attacks, and therefore virtually the only approach successful in protecting privacy in this context. In essence, differential privacy consists of subtly adding noise to the data such that individual privacy is protected while meaningful statistical inference still remains possible. A problem here is that it is not always known beforehand which specific statistical fact will be considered interesting.

Clearly it is impossible to have perfect privacy ("publish nothing") and perfect statistical inference ("publish everything") at the same, so differential privacy offers a calibrated trade-off between the two extremes. Indeed, adding noise to perfect, pristine data may seem like a no-brainer, but it is the only way to preserve a minimum of privacy in a data-driven world. Giving up on privacy is not an option, so choices will have to be made. Society needs to become aware of this dilemma and discuss the pros and cons of the various solutions.

The ADCS domain administrator is right there

Fabricio Gimenes

Offensive Security Specialist, Telefônica Brasi

What is ADCS?

Active Directory Certificate Services (AD CS)

ADCS provides customizable services for issuing and managing public key infrastructure (PKI) certificates used in software security systems that employ public key technologies. The digital certificates that AD CS provides can be used to encrypt and digitally sign electronic documents and messages. Further, these digital certificates can be used for authentication of the computer, user or device accounts on a network. Digital certificates are used to provide:

- 1) Confidentiality - through encryption
- 2) Integrity - through digital signatures
- 3) Authentication - by associating certificate keys with the computer, user, or device accounts on a computer network.

What are Templates?

Templates

AD CS Enterprise CAs issue certificates with settings defined by AD objects known as certificate templates. These templates are collections of enrollment policies and predefined certificate settings and contain things like "How long is this certificate valid for?", "What is the certificate used for?", "How is the subject specified?", "Who is allowed to request a certificate?", and a myriad of other settings:

These certificate services were available starting in Windows 2000 and continue to be available as a server role in Windows Server 2008 R2.

Misconfigured Certificate Templates — ESC1

In order to abuse this misconfiguration, the following conditions must be met:

- 1) The Enterprise CA grants low-privileged users enrollment rights.
- 2) Manager approval is disabled.
- 3) No authorized signatures are required.

NTLM Relay to AD CS HTTP Endpoints — ESC8

In this attack it is possible to obtain domain admin over HTTP Based Authentication. As covered in the “Certificate Enrollment” section, AD CS supports several HTTP-based enrollment methods via additional AD CS server roles that administrators can install. These HTTPbased certificate enrollment interfaces are all vulnerable NTLM relay attacks. Using NTLM relay, an attacker on a compromised machine can impersonate any inbound-NTLM-authenticating AD account. While impersonating the victim account, an attacker could access these web interfaces and request a client authentication certificate based on the User or Machine certificate templates.

And for end point over this attack that explain here is we can obtain total control on ADCS

CA Manager - ESC7

When a user has the Manage CA or Manage Certificates access right on a CA. While there are no public techniques that can abuse only the Manage Certificates access right for domain privilege escalation, we can still use it to issue or deny pending certificate requests.

The Joy of Exploiting the Kernel on 2022

Eduardo Vela

Product Security Response TL/M, Google

During 2022 we received as part of kCTF dozens of exploits for several vulnerabilities in the Linux Kernel. We spent a lot of time analyzing them and trying to learn from them. In this talk we will present the best and worst lessons to take from these vulnerabilities and exploits and teach the audience how to bake a perfect root shell with the techniques we saw so far with kCTF exploits. No previous knowledge of Kernel security is necessary, just familiarity with the basics of memory corruption vulnerabilities (buffer overflow, use-after-free, double free).

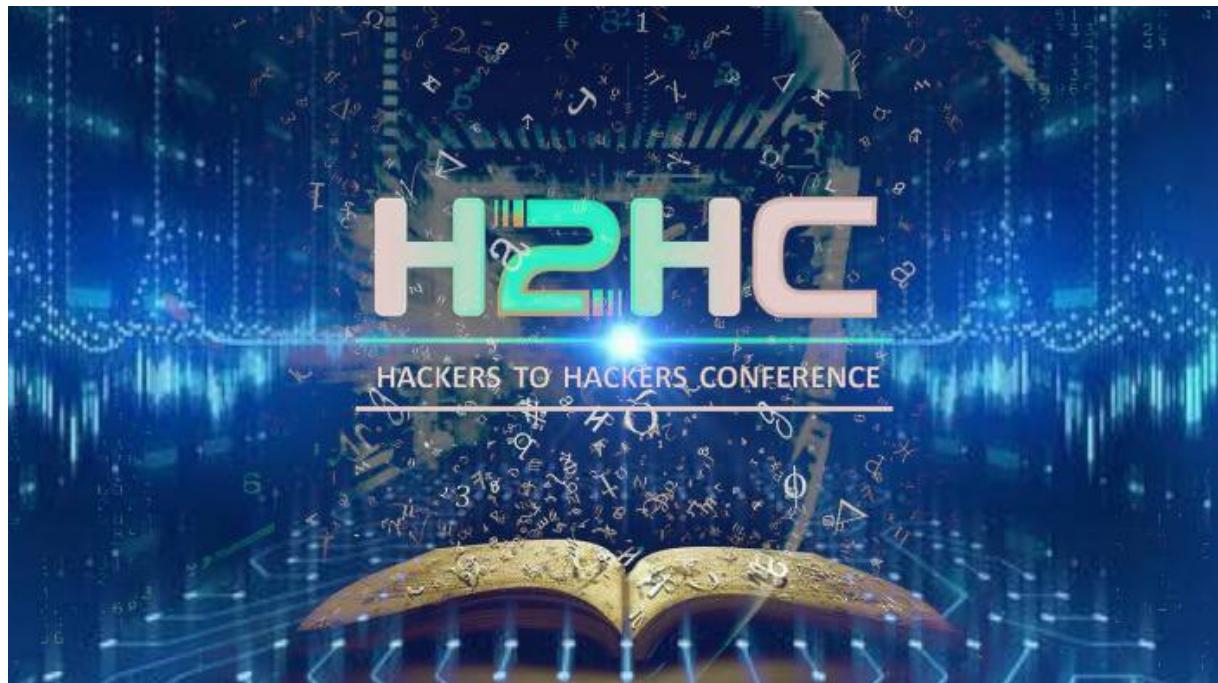
To branch or not to branch: security implications of x86 frontend implementations

Pawel Wieczorkiewicz

Security Researcher, Open Source Security Inc

In this talk, we discuss a flaw recently discovered in AMD x86 processors of various microarchitectures: Zen1, Zen2 and Zen3, and its role in a speculative execution vulnerability type called straight-line speculation (SLS). We begin with a brief overview of the AMD BPU specification, focusing on its sub-components

involved in branch prediction of direct unconditional and conditional branches. Next, we discuss direct conditional branches misprediction and methods to reliably achieve it across privilege boundaries or cross hyper-threads, followed by a discussion of the resulting speculation window and its potential to create exploitable Spectre v1 gadgets. We also demonstrate why Spectre v1 gadgets are not limited to array out-of-bound access and memory access latency related speculation. Next, we present details of a new and surprising vulnerability of some AMD processors: direct unconditional branch SLS (CVE-2021-26341). After a quick introduction to the SLS topic, we analyze the resulting speculation window, cross hyper-threads influence and potential ways of finding and exploiting the unexpected SLS gadgets. Finally, we take a quick survey over proposed mitigations for the vulnerabilities in direct unconditional and conditional branches speculation.



Palestrantes

Alex Ionescu

Technical Director of Platform Operations, CSE (Communications Security Establishment)

Alex Ionescu is the Technical Director, Platform Operations and Research at CSE (Communications Security Establishment), Canada's National Cryptologic Agency. Previously, he was the VP of Endpoint Engineering at CrowdStrike, Inc., where he started as the Founding Chief Architect in 2011. Alex is a world-class security architect and consultant expert in low-level system software, kernel development, security training, and reverse engineering. He is co-author of the last 3 editions of the Windows Internals series. During the last two decades, his work led to the fixing of dozens of critical kernel vulnerabilities in Windows. Previously, Alex was the lead kernel developer for ReactOS, an open source Windows clone written from scratch, for which he wrote most of the Windows NT-based subsystems. During his studies in Computer Science, Alex worked at Apple on the iOS kernel, boot loader, and drivers on the original core platform team behind the iPhone, iPad, and AppleTV. Alex is also the founder of Winsider Seminars & Solutions Inc., a company that specializes in low-level system software, reverse engineering and security training for various institutions.

Keynote: "Keynote by Alex Ionescu"

Alexander Ermolov

Principal Security Researcher, Binarly Inc.

Alex Ermolov leads supply chain and platform security research and development at Binarly Inc. With more than 10 years of experience in researching low-level design, firmware and systems software built for various platforms and architectures, he helps to create a solution for protecting devices against firmware threats.

Palestras: "Data only Attacks Against UEFI BIOS" e "Breaking Firmware Trust from Pre EFI: Exploiting Early Boot Phases"

Alexandra Sandulescu

Security Engineer, Google

My name is Alexandra Sandulescu and for the past 5 years I have been working on various security research topics from fuzzing to speculative execution attacks to sandboxing. Currently I am a Security Engineer at Google.

Palestra: "A researcher's take on Spectre exploits"

Brian Butterly

Security Engineer

After a few years of incident response in a very large and crazily diverse environment, Brian has changed back into a more offensive area. Focusing on operational technology and the railway sector, he's applying his knowledge from past projects in the areas of embedded-, hardware-, mobile- and telecommunications-security to ginormous vehicles driving at high speeds and everything surrounding them. While combining a closed environment and good old hacking spirit results in a fair amount of challenges, he's doing his best to fuse both world together and carry on sharing fun insights.

Palestra: "An Insight into Railway Security"

Bruno Cortes

Lead Offensive Security Engineer, Nubank

Professional with more than 10 years of career in Cyber Security, working at companies in the Technology and Military segments, such as: Brazilian Army, GoHacking and Nubank.

Palestra: "From PIX to Reverse Shell"

Bruno Macabeus

Sr. Full-Stack Engineer, Anima

Hello, I'm Macabeus!

I'm a Sr. Full-Stack Engineer at Anima, a startup company that improves the experience for designers and front-end developers by automating their workflows. I work building plugins for design tools such as Figma and Adobe XD, as well as coding on our Python/Node backend.

I learned how to code by contributing to open-source projects, so I love helping them in different ways. I have several personal projects on GitHub, approaching different topics, including reverse engineering, compilers, and VS Code plugins.

Last but not least, I love developer communities, and I have been helping many of them for years now, both as an organizer and speaker.

Check more about me and my projects on my site: <https://macabeus.github.io/>

Palestra: "Engenharia Reversa num jogo de Gameboy Advance"

Edmond Rogers

Academic Researcher

Before joining the University of Illinois Information Trust Institute (ITI) in 2011, Edmond Rogers was actively involved as an industry participant in many research activities in ITI's TCIPG Center, including work on CyPSA Cyber Physical Situational Awareness, NetAPT (the Network Access Policy Tool) and LZFuzz (Proprietary Protocol Fuzzing). Rogers also has developed and delivers customized training on ICS defense at the TCIPG Summer School and to utilities directly. Rogers leverages his wealth of experience to assist ITI researchers in creating laboratory conditions that closely reflect real-world configurations. Rogers has spoken across the world regarding defense of critical infrastructure at conferences such as Bsides London, H2HC, Black Hat, Defcon, BsidesLV, Troopres, BerlinSides and he is currently the president of Hackito Ergo Sum.

Palestra: "Exploitation tactics discovery using data analytics"

Eduardo Vela

Product Security Response TL/M, Google

Eduardo has been cooking vulnerabilities for almost 2 decades, which means he is getting older and older. His love for penguin vulnerabilities started after working on kCTF. He now spends time working on kernel exploit cooking recipes and producing the videos and recipe books for them. He currently does vulnerability stuff at Google and working with the security community to find and exploit all types of vulnz. His lifelong dream is to work at McDonald's. He is a terrible cook.

Palestra: "The Joy of Exploiting the Kernel on 2022"

Evandro Hora

Socio-Fundador e Diretor da Tempest Security Intelligence

Keynote: "Ensaio sobre o Recrutamento"

Fabricio Gimenes

Offensive Security Specialist, Telefônica Brasil

My name is Fabricio Gimenes, I'm 36 years old. I have worked with information security for about 13 years, I worked in big companies like "Banks, Ensures, E-commerce, etc". I currently work as an offensive security specialist at Telefônica Brasil.

Palestra: "The ADCS domain administrator is right there"

Fernando Gont

Security Consultant & Researcher, SI6 Networks

Fernando Gont has twenty years of industry experience in the fields of Internet engineering and information security, working for both private and governmental organizations from around the world.

He has authored more than 35 Internet Engineering Task Force (IETF) RFCs (many of which focusing on IPv6 security), and has also produced the SI6 Networks' IPv6 Toolkit (a security assessment toolkit for the IPv6 protocol suite).

More information about Fernando Gont is available at his personal web site: <<https://www.gont.com.ar>>

Palestra: "State of the Art in IPv6 Attack & Defense"

Fernando Merces

Senior Threat Researcher, Trend Micro

Fernando é Pesquisador de Ameaças na Trend Micro (<https://www.trendmicro.com/>), onde atua como investigador de ciber crime, utilizando engenharia reversa e técnicas de inteligência de ameaças no time de Pesquisa de Ameaças Futuras (FTR). Criador de várias ferramentas livres (<https://github.com/merces>) na área, com frequência apresenta suas pesquisas nos principais eventos de segurança no Brasil e no exterior. É também professor e fundador da Mente Binária (<https://www.mentebinaria.com.br>), uma instituição de ensino e pesquisa sem fins lucrativos comprometida com o ensino de computação no Brasil.

Palestra: "Dont Blink! A deep dive into Cyclops Blink"

Gustavo Scotti

Secure Firmware Engineer, Microsoft Corporation

Gustavo Scotti (a.k.a. csh) writes secure firmware to security processors at Microsoft. An old-school hacker who wrote a few exploits, hacked PlayStations, secured Xboxes, and broke a few security systems (hardware and software).

Palestra: "Eleicoes Transparentes"

Jeroen van de Graaf

Professor, Universidade Federal de Minas Gerais

I am a cryptographer focussing on the theoretical and the applied aspects of cryptographic protocols.

I am particularly interested in unconditional privacy, including topics such as: quantum cryptography; protocols based on noise; authentication based on signal reconciliation; election and mixing protocols; general two- and multi-party computation; privacy-preserving data mining etc. Many of these protocol borrow techniques from information theory and coding theory. I have a Master's in mathematics from the Universiteit van Amsterdam (1985) and a PhD in Informatics from the Université de Montréal (1998). From August 2008 until February 2011 I was an Assistant Professor at the Universidade Federal de Ouro Preto. Since March 2011 I moved to the Universidade Federal de Minas Gerais.

Palestra: "Techniques for publishing statistical information without violating individual's privacy according to the LGPD and GDPR"

Johannes Wikner

PhD student, ETH Zurich

Johannes Wikner is a PhD student at COMSEC, a research group at ETH Zurich that does security research at the lower levels of the computing stack, including the hardware. His research concerns microarchitectural security of closed source commodity hardware, where he makes CPUs misbehave for fun and profit (and for science!).

Palestra: "Retbleed: Arbitrary Speculative Code Execution with Return Instructions"

Jorge Buzeti

Cyber Security Analyst, ISH

Jorge é um jovem de 16 anos apaixonado por hacking, principalmente em áreas de low-level entre programação em C e pwn, trabalha atualmente na ISH como Cyber Security Analyst, é membro fundador da HardDisk(<https://harddisk.com.br>) e atua em projetos open-source como o brokepkg, rookit para o Linux.

Palestra: "Browser Exploitation and JIT Compilers"

Lourenco A. Pereira

Professor, Instituto Tecnológico de Aeronáutica(ITA)

Doutor (2016) e Mestre (2010) em Ciências de Computação e Matemática Computacional pela Universidade de São Paulo, Bacharel em Ciência da Computação pela Universidade de Alfenas (2006). Atualmente Professor no Instituto Tecnológico de Aeronáutica - ITA nas áreas de Segurança Cibernética e Redes de Computadores. Foco no estudo de sistemas operacionais e implementações de pilhas de protocolos de redes. Atualmente realizando pesquisas em segurança cibernética em infraestruturas críticas que envolvem 5G/6G, Internet das Coisas, Sistemas de Transportes Inteligentes e malwares. Tópicos de interesse: detecção de tráfego malicioso por meio de assinaturas e anomalias para desenvolvimento de

nova geração de sistemas de detecção e prevenção de intruções; implementação de mecanismo de proteção (firewall) baseado em zero-trust; enumeração e exploração de firmware em grande escala; exploração de tecnologias de redes veiculares (WAVE e 5G); e orquestração de cadeia de comando e controle de malware. Responsável pelo Laboratório de Comando & Controle e Defesa Cibernética do ITA, onde conduz projetos de científicos e orienta alunos de doutorado, mestrado, e iniciação científica.

Instituto Tecnológico de Aeronáutica
Membro Efetivo da Sociedade Brasileira de Computação e da IEEE

Palestra: "Aprendizado de Máquina em Segurança"

Magno Logan

Information Security Specialist, Trend Micro

Magno Logan works as an Information Security Specialist for Trend Micro. He specializes in Cloud, Container, and Application Security Research, Threat Modelling, and DevSecOps. In addition, he has been tapped as a resource speaker for numerous security conferences around the globe.

Palestra: "GitHub Actions: Vulnerabilities, Attacks and Counter Measures"

Matrosov & Ermolov & Vasilenko & Thomas

Binarly Inc.

Alex Matrosov is CEO and Founder of Binarly Inc. where he builds an AI-powered platform to protect devices against emerging firmware threats. Alex has more than two decades of experience with reverse engineering, advanced malware analysis, firmware security, and exploitation techniques. He served as Chief Offensive Security Researcher at Nvidia and Intel Security Center of Excellence (SeCoE). Alex is the author of numerous research papers and the bestselling award-winning book Rootkits and Bootkits: Reversing Modern Malware and Next Generation Threats. He is a frequently invited speaker at security conferences, such as RECon, Black Hat, Offensivecon, WOOT, DEF CON, and many others. Additionally, he was awarded multiple times by Hex-Rays for his open-source contributions to the research community.

Alex Ermolov - Ver palestrante "Alexander Ermolov"

Yegor Vasilenko is an experienced Security Researcher focused on reverse engineering and firmware analysis. Nowadays he enjoys firmware reverse engineering and tools development. Yegor is one of the maintainers of a popular tool called efiXplorer for UEFI firmware reverse engineering and vulnerability research.

Dr. Sam L. Thomas is a security researcher and former academic from the UK. His interests include reverse engineering, malware detection, and static analysis. Before leaving academia, he completed post-docs in France (at CNRS) and the UK (at the University of Birmingham) and was Maître de conférences at CentraleSupélec, France. His PhD thesis focused on devising novel approaches to detect backdoors

in embedded device firmware. He has presented his research at numerous internationally renowned academic conferences, including CHES, RAID, ESORICS, and DIMVA. He has also served on the program committees for DIMVA (2019-2022) and WOOT (2019, 2020).

Palestra: "Breaking Firmware Trust from Pre EFI: Exploiting Early Boot Phases"

Pawel Wieczorkiewicz

Security Researcher, Open Source Security Inc

Pawel Wieczorkiewicz is a Security Researcher at Open Source Security Inc., a company developing the state-of-the-art Linux kernel hardening solution known as grsecurity. His research focuses on offensive security aspects of transient and speculative execution vulnerabilities, side-channels, and the effectiveness of defensive mitigations in OSes and hypervisors. Pawel's deep interest in low-level security of software and hardware has resulted in the discovery of a number of vulnerabilities in AMD and Intel processors in addition to the Linux kernel and Xen hypervisor system software.

Palestra: "To branch or not to branch: security implications of x86 frontend implementations"

pirata & belphegor

Darkweb, Worldwide

pirata & belphegor are friends and have been working together for more than a decade. They constantly try to find new challenges that have nothing to do with their direct work, as a way to prevent boredom. They can be found in different corners of the underweb criticizing big techs and developing new technologies that could otherwise become startups but instead are provided to the community for free.

Palestra: "Desenvolvimento de um servico de email privativo"

Tarakanov & Labunets

Independent Security Researchers

Nikita Tarakanov is an independent security researcher. He has worked as a security researcher in Positive Technologies, Vupen Security, Intel Corporation and Huawei. He likes writing exploits, especially for OS kernels. He won the PHDays Hack2Own contests in 2011 and 2012. He has published a few papers about kernel mode drivers and their exploitation. He is currently engaged in reverse engineering research and vulnerability search automation.

Andrey Labunets is a security researcher with more than a decade of experience in vulnerability research and reverse engineering.

Palestra: "Cinema Time!"

Artigo | NUCLEI - Scan rápido e versátil

Autor: Marcos Rosário (god6ixx)

Registro Único de Artigo

<https://doi.org/10.47986/16/1>

Resumo

Dentre os pesquisadores, bug hunters e profissionais da área de Cyber Security, há quem ainda use apenas NMAP [1] e Burp Suite [2] para identificação de vulnerabilidades em aplicações. Meu intuito neste breve artigo é apresentar uma nova possibilidade que facilite a vida de meus colegas e dos leitores aqui da H2HC Magazine.

Introdução

NUCLEI [3] é uma ferramenta baseada na linguagem GO [4] que utiliza o conceito de templates em YAML [5] para enviar e processar as requisições. Suas vantagens incluem a identificação de vulnerabilidades com minimização de falso positivos, alta diversidade de categorias de templates e a possibilidade de criar o seu próprio template, seja para alguma falha que ainda não há um modelo¹ disponível ou para alguma CVE de sua propriedade.

Instalação e Execução

Como dito anteriormente, a ferramenta é baseada na linguagem da “marmota azul” e você pode seguir o processo de instalação disponível no repositório oficial [3] ou, se preferir, nas versões mais recentes foi disponibilizada a instalação por meio do gerenciador de pacotes Linux APT.

Há diversas possibilidades para a sua execução, as mais comuns e que iremos utilizar ao longo deste template são:

- Para um alvo e template específicos:
\$ nuclei -u [URL/IP] -t [Template desejado]
- Para utilizar todos os templates disponíveis na ferramenta em um alvo específico:
\$ nuclei -l [PATH do arquivo com os hosts]

¹Em alguns momentos preferi utilizar a palavra "modelo" no lugar de "template" ao longo do texto para não tornar a leitura cansativa ou repetitiva.

Templates

O template é um modelo que define como a ferramenta deve realizar as validações necessárias em seu alvo a fim de obter o resultado desejado na detecção de vulnerabilidades.

Por padrão, ao instalar o NUCLEI em sua máquina ou VPS, você instala também um pacote de templates de diversas categorias. Há templates disponíveis para diversas vulnerabilidades já conhecidas de serviços, servidores, CMSs e etc. Dentre eles estão: Jenkins [6], Drupal [7], Grafana [8], Wordpress [9], Confluence [10], GitLab [11], entre outros. É importante destacar a variedade de modelos disponíveis, sendo 4023 templates (na versão atual - 9.2.1), como os exemplos da Figura 1.

```
+ nuclei-templates ls
CODE_OF_CONDUCT.md      README_KR.md          cndv           dns           fuzzing        miscellaneous
CONTRIBUTING.md        TEMPLATES-STATS.json    contributors.json   exposed-panels   headless       misconfiguration
LICENSE.md              TEMPLATES-STATS.md      cves           exposures     helpers        network
PULL_REQUEST_TEMPLATE.md TOP-10.md            default-logins   file          iot           ssl
+ nuclei-templates tree vulnerabilities
vulnerabilities
├── apache
│   ├── apache-flink-unauth-rce.yaml
│   ├── apache-ofbiz-log4j-rce.yaml
│   ├── apache-solr-file-read.yaml
│   └── apache-solr-log4j-rce.yaml
├── backdoor
│   └── jboss-backdoor.yaml
├── cisco
│   ├── cisco-unified-communications-log4j.yaml
│   ├── cisco-vmanage-log4j.yaml
│   └── cuam-username-enumeration.yaml
├── code42
│   └── code42-log4j-rce.yaml
├── concrete
│   └── concrete-xss.yaml
├── confluence
│   └── confluence-ssrf-sharelinks.yaml
├── dedecms
│   ├── dedecms-carbuyaction-fileinclude.yaml
│   ├── dedecms-config-xss.yaml
│   ├── dedecms-membergroup-sqli.yaml
│   ├── dedecms-openredirect.yaml
│   └── dedecms-rce.yaml
├── drupal
│   └── drupal-avatar-xss.yaml
├── fastjson
│   ├── fastjson-1-2-24-rce.yaml
│   ├── fastjson-1-2-41-rce.yaml
│   ├── fastjson-1-2-42-rce.yaml
│   ├── fastjson-1-2-43-rce.yaml
│   ├── fastjson-1-2-47-rce.yaml
│   ├── fastjson-1-2-62-rce.yaml
│   ├── fastjson-1-2-67-rce.yaml
│   └── fastjson-1-2-68-rce.yaml
└── generic
    ├── basic-xss-prober.yaml
    ├── cache-poisoning.yaml
    ├── cors-misconfig.yaml
    ├── crlf-injection.yaml
    ├── error-based-sql-injection.yaml
    ├── generic-blind-xxe.yaml
    ├── generic-j2ee-lfi.yaml
    ├── generic-linux-lfi.yaml
    ├── generic-windows-lfi.yaml
    └── host-header-injection.yaml
```

Figura 1: Lista de pastas de templates disponíveis por padrão na ferramenta

Além da análise de vulnerabilidades web conhecidas e redução de falso positivos (o que por si já é vantajoso), a ferramenta também inclui scan de redes, enumeração de tecnologias utilizadas pela aplicação, workflow de endpoints em APIs e etc. No exemplo das Figuras 2 e 3 subi uma máquina vulnerável onde podemos observar que diversos pontos de entrada foram identificados utilizando apenas alguns templates, como os relacionados a network e misconfiguration.

```
+ nuclei-templates nuclei -u http://localhost/dvwa/vulnerabilities/ -t network

____ _ _ ____/_/ _ ( )
 / __ \ \ / / _ / _ \ \ \ / 
/ / / / / / / / _ / / _ / / 
/_ / _ \_, _ \_ / _ \_ / _ /  2.7.7

projectdiscovery.io

[WRN] Use with caution. You are responsible for your actions.
[WRN] Developers assume no liability and are not responsible for any misuse or damage.
[INF] Using Nuclei Engine 2.7.7 (latest)
[INF] Using Nuclei Templates 9.2.1 (latest)
[INF] Templates added in last update: 0
[INF] Templates loaded for scan: 38
[2022-09-26 06:32:54] [mysql-native-password] [network] [info] localhost:3306
[2022-09-26 06:32:54] [openssh-detection] [network] [info] localhost:22 [SSH-2.0-OpenSSH_8.6]
```

Figura 2: Utilizando os templates relacionados a vulnerabilidades de redes

```
+ nuclei-templates nuclei -u http://localhost/dvwa/vulnerabilities/ -t misconfiguration

____ _ _ ____/_/ _ ( )
 / __ \ \ / / _ / _ \ \ \ / 
/ / / / / / / / _ / / _ / / 
/_ / _ \_, _ \_ / _ \_ / _ /  2.7.7

projectdiscovery.io

[WRN] Use with caution. You are responsible for your actions.
[WRN] Developers assume no liability and are not responsible for any misuse or damage.
[INF] Using Nuclei Engine 2.7.7 (latest)
[INF] Using Nuclei Templates 9.2.1 (latest)
[INF] Templates added in last update: 0
[INF] Templates loaded for scan: 236
[INF] Templates clustered: 34 (Reduced 30 HTTP Requests)
[INF] Using Interactsh Server: oast.me
[2022-09-26 06:34:14] [http-missing-security-headers:x-content-type-options] [http] [info] http://localhost/dvwa/vulnerabilities/
[2022-09-26 06:34:14] [http-missing-security-headers:clear-site-data] [http] [info] http://localhost/dvwa/vulnerabilities/
[2022-09-26 06:34:14] [http-missing-security-headers:access-control-max-age] [http] [info] http://localhost/dvwa/vulnerabilities/
[2022-09-26 06:34:14] [http-missing-security-headers:access-control-allow-methods] [http] [info] http://localhost/dvwa/vulnerabilities/
[2022-09-26 06:34:14] [http-missing-security-headers:access-control-allow-headers] [http] [info] http://localhost/dvwa/vulnerabilities/
[2022-09-26 06:34:14] [http-missing-security-headers:strict-transport-security] [http] [info] http://localhost/dvwa/vulnerabilities/
[2022-09-26 06:34:14] [http-missing-security-headers:x-permitted-cross-domain-policies] [http] [info] http://localhost/dvwa/vulnerabilities/
[2022-09-26 06:34:14] [http-missing-security-headers:cross-origin-embedder-policy] [http] [info] http://localhost/dvwa/vulnerabilities/
[2022-09-26 06:34:14] [http-missing-security-headers:access-control-allow-origin] [http] [info] http://localhost/dvwa/vulnerabilities/
[2022-09-26 06:34:14] [http-missing-security-headers:access-control-expose-headers] [http] [info] http://localhost/dvwa/vulnerabilities/
[2022-09-26 06:34:14] [http-missing-security-headers:permission-policy] [http] [info] http://localhost/dvwa/vulnerabilities/
[2022-09-26 06:34:14] [http-missing-security-headers:referrer-policy] [http] [info] http://localhost/dvwa/vulnerabilities/
[2022-09-26 06:34:14] [http-missing-security-headers:cross-origin-opener-policy] [http] [info] http://localhost/dvwa/vulnerabilities/
[2022-09-26 06:34:14] [http-missing-security-headers:content-security-policy] [http] [info] http://localhost/dvwa/vulnerabilities/
[2022-09-26 06:34:14] [http-missing-security-headers:x-frame-options] [http] [info] http://localhost/dvwa/vulnerabilities/
[2022-09-26 06:34:14] [http-missing-security-headers:cross-origin-resource-policy] [http] [info] http://localhost/dvwa/vulnerabilities/
[2022-09-26 06:34:14] [http-missing-security-headers:access-control-allow-credentials] [http] [info] http://localhost/dvwa/vulnerabilities/
```

Figura 3: Utilizando os templates relacionados a misconfiguration

Para minimizar falso positivos, a ferramenta utiliza um padrão de arquivo YAML com as várias possibilidades e validações que ela precisa para aumentar a especificidade como, por exemplo, verificar se algum cabeçalho encontra-se ausente ou se um determinado serviço está ativo em sua respectiva porta. Com isto, é possível também que você crie o seu próprio template [12] para identificação de vulnerabilidades em

que você é o proprietário da CVE ou que ainda não esteja disponível num modelo dentre os já disponíveis da ferramenta. A Figura 4 ilustra como um template pode minimizar falso positivos.

```

1 id: xss-deprecated-header-detect
2
3 info:
4   name: XSS-Protection Header - Cross-Site Scripting
5   author: joshlarsen
6   severity: info
7   description: XSS-Protection header in Explorer, Chrome, and Safari contains a cross-site scripting vulnerability if set to any value other than '0'.
8   reference:
9     - https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-XSS-Protection
10    - https://owasp.org/www-project-secure-headers/#x-xss-protection
11 classification:
12   cvss-metrics: CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:C/L/I:L/A:N
13   cvss-score: 7.2
14   cwe-id: CWE-79
15   tags: xss,misconfig,generic
16
17 requests:
18   - method: GET
19     path:
20       - "{{BaseUrl}}"
21
22 matchers-condition: and
23 matchers:
24
25   - type: regex
26     part: header
27     regex:
28       - "(?i)x-xss-protection: 0"
29     negative: true
30
31   - type: regex
32     part: header
33     regex:
34       - "(?i)x-xss-protection: 1+"
35
36 extractors:
37   - type: kval
38     part: header
39     kval:
40       - x_xss_protection
41
42 # Enhanced by mp on 2022/09/15

```

INFORMAÇÕES RELACIONADAS AO
TEMPLATE E A VULNERABILIDADE

DADOS UTILIZADOS PELA FERRAMENTA
PARA VALIDAR SE A FALHA DE FATO É EXISTENTE

Figura 4: Estrutura de um template utilizado pela ferramenta

Os templates são o grande diferencial desta tool² e você pode turbiná-la ainda mais, adicionando modelos criados pela comunidade nacional e internacional. Aqui vão algumas fontes de templates interessantes para incluir em seu arsenal:

- <https://github.com/z3bd/nuclei-templates>
- <https://github.com/osintbrazuca/nuclei-templates-brazuca>
- <https://github.com/Harish4948/Nuclei-Templates>
- <https://github.com/randomstring/nuclei-sap-templates>
- <https://github.com/sadnansakin/my-nuclei-templates>
- <https://github.com/redteambrasil/nuclei-templates>
- <https://github.com/emadshanab/Nuclei-Templates-Collection>

²Em alguns momentos preferi utilizar a palavra "tool" no lugar de "ferramenta" ao longo do texto para não tornar a leitura cansativa ou repetitiva.

Cenário real

Em Pentests e Bug Bounties, o NUCLEI facilita a identificação de possíveis vulnerabilidades e disponibiliza a criação de seu próprio template para buscar aquela falha específica que seria mais crítica ou que você percebe que há um padrão para identificá-la. É possível também utilizar a tool em outros cenários, como na validação de vulnerabilidades em code-review.

Como por exemplo, em cenários reais, onde executo em maior parte pentests mobile, a análise estática é uma fase que deve ser realizada em todas as ocasiões. A validação de permissões, chaves de API expostas e informações que possam causar algum impacto podem ser analisadas no código-fonte da aplicação. Para demonstrar a automatização deste processo, criaremos a seguir um template para auxiliar nas verificações.

Usando de exemplo a aplicação DIVA (Damn insecure and vulnerable App) [13], realizei a engenharia reversa do APK (Android Application Pack - formato de arquivo destinado ao sistema operacional Android) e, além do uso de alguns templates já prontos que auxiliam na fase da análise estática, desenvolvi o modelo YAML presente na Figura 5 para analisar se a aplicação contém a flag "allowBackup"³ ativa.



```
god6ixx's Terminal (vim) #1
1 id: flag-backup-validation
2
3 info:
4   name: Flag Backup Validation
5   author: god6ixx
6   severity: low
7
8 file:
9   - extensions:
10     - all
11
12   matchers:
13     - type: word
14       words:
15         - "android:allowBackup=\"true\""
16
~
```

god6ixx's Terminal (-zsh) #2

Figura 5: Template desenvolvido para validar a flag “allowBackup” nos arquivos do Android

Normalmente, faríamos a validação manualmente no arquivo "AndroidManifest.xml", que é responsável por indicar ao sistema operacional as permissões e configurações em que a aplicação deve ser executada, mas, nesse caso, a ferramenta NUCLEI será utilizada para automatizar tal tarefa. Como podemos observar na Figura 6, utilizei a ferramenta nos arquivos estruturais da aplicação, obtidos por meio da etapa de engenharia reversa citada anteriormente. Além de outras falhas obtidas pelos templates já prontos, também foi identificado pelo template da Figura 5 que a flag citada encontra-se ativa.

³Caso a flag “allowBackup” esteja definida como “true”, um invasor ou usuário malicioso pode realizar backup dos arquivos internos da aplicação de forma arbitrária utilizando sistemas de backup como “adb backup” [14] e a funcionalidade de backup padrão do Android.

```
+ ➤ diva echo diva-beta/ | nuclei -t Android  
____/ /_ ( )  
/_ \ // _/ -\ \/  
/_/_/_/_/_/_/_/_/_/  
/_/_\_,_\_,_\_,_\_ 2.7.7  
  
projectdiscovery.io  
  
[WRN] Use with caution. You are responsible for your actions.  
[WRN] Developers assume no liability and are not responsible for any misuse or damage.  
[INF] Using Nuclei Engine 2.7.7 (latest)  
[INF] Using Nuclei Templates 9.2.1 (latest)  
[INF] Templates added in last update: 0  
[INF] Templates loaded for scan: 42  
[2022-09-30 13:33:26] [flag-backup-validation] [file] [low] diva-beta/AndroidManifest.xml  
[2022-09-30 13:33:26] [android-debug-enabled] [file] [low] diva-beta/AndroidManifest.xml  
[2022-09-30 13:33:30] [dynamic-registered-broadcast-receiver] [file] [info] diva-beta/smali/android/support/v4/media/TransportMediatorJellybeanMR2.smali  
[2022-09-30 13:33:39] [webview-javascript-enabled] [file] [info] diva-beta/smali/jakhar/aseem/diva/InputValidation2URISchemeActivity.smali  
[2022-09-30 13:33:39] [webview-load-url] [file] [info] diva-beta/smali/jakhar/aseem/diva/InputValidation2URISchemeActivity.smali
```

Figura 6: Resultado da análise realizada pela ferramenta com base nos templates

Conclusão

Esta é uma daquelas ferramentas que não pode faltar no arsenal, pois além da customização excelente e de sua alta versatilidade, os scans são rápidos devido a quantidade de testes realizados, que neste caso são bem mais assertivos. Ela pode ser melhor utilizada quando em conjunto com outras ferramentas, como Subfinder [15], HTTPX [16], GAU [17] e KXSS [18].

Vale ressaltar que apesar deste tipo de ferramenta facilitar a análise e acelerar o processo de identificação e validação das vulnerabilidades, não podemos nos limitar apenas ao seu uso, afinal, hacker hackeia. :)

Referências

- [1] “NMAP,” acessado em 01-Outubro-2022. [Online]. Disponível em: <https://nmap.org/>
 - [2] “Burp Suite,” acessado em 01-Outubro-2022. [Online]. Disponível em: <https://portswigger.net/burp>
 - [3] “NUCLEI,” acessado em 01-Outubro-2022. [Online]. Disponível em: <https://github.com/projectdiscovery/nuclei>
 - [4] “GO,” acessado em 01-Outubro-2022. [Online]. Disponível em: <https://go.dev/>
 - [5] “YAML,” acessado em 01-Outubro-2022. [Online]. Disponível em: <https://yaml.org/>
 - [6] “Jenkins,” acessado em 01-Outubro-2022. [Online]. Disponível em: <https://www.jenkins.io/>
 - [7] “Drupal,” acessado em 01-Outubro-2022. [Online]. Disponível em: <https://www.drupal.org/>
 - [8] “Grafana,” acessado em 01-Outubro-2022. [Online]. Disponível em: <https://grafana.com/>
 - [9] “Wordpress,” acessado em 01-Outubro-2022. [Online]. Disponível em: <https://wordpress.com/>

- [10] "Confluence," acessado em 01-Outubro-2022. [Online]. Disponível em: <https://www.atlassian.com/software/confluence>
- [11] "GitLab," acessado em 01-Outubro-2022. [Online]. Disponível em: <https://about.gitlab.com/>
- [12] "NUCLEI Templating Guide," acessado em 01-Outubro-2022. [Online]. Disponível em: <https://nuclei.projectdiscovery.io/templating-guide>
- [13] "DIVA Android," acessado em 01-Outubro-2022. [Online]. Disponível em: <https://github.com/payatu/diva-android>
- [14] "Android Debug Bridge (adb)," acessado em 01-Outubro-2022. [Online]. Disponível em: <https://developer.android.com/studio/command-line/adb>
- [15] "Subfinder," acessado em 01-Outubro-2022. [Online]. Disponível em: <https://github.com/projectdiscovery/subfinder>
- [16] "HTTPX," acessado em 01-Outubro-2022. [Online]. Disponível em: <https://github.com/projectdiscovery/httpx>
- [17] "GAU," acessado em 01-Outubro-2022. [Online]. Disponível em: <https://github.com/lc/gau>
- [18] "KXSS," acessado em 01-Outubro-2022. [Online]. Disponível em: <https://github.com/tomnomnom/hacks/tree/master/kxss>

Marcos Rosário (god6ixx) - god6ixxo@gmail.com

Marcos Rosário, graduado em Defesa Cibernética pela FIAP, possui mais de 4 anos de experiência em segurança ofensiva com especialização em testes de intrusão nas categorias: Web, Mobile, API, Infraestrutura e Wireless.

Ministrou palestras relacionadas a segurança da informação na OSINT Village e FIAP. Co-fundador da comunidade hacker "Boitat-ech", que tem como intuito auxiliar jovens a ingressarem na área de segurança da informação e difundir conhecimento sobre Ethical Hacking.

Atua hoje na PRIDE Security como Information Security Consultant.



Artigo | Web Cache Deception Attack

Autor: Maycon Vitali (OutOfBound)

Registro Único de Artigo

<https://doi.org/10.47986/16/2>

Introdução

Este artigo tem por finalidade apresentar uma classe de ataque não muito nova (apresentada inicialmente em 2017 por Omer Gil [1] [2]) mas que, apesar de simples, é pouco explorada no cotidiano. Estamos falando do Web Cache Deception Attack, um ataque que tem por finalidade principal obter dados sensíveis de usuários legítimos de uma determinada aplicação afetada.

Será apresentado uma fundamentação básica sobre serviços de cache, cujo conhecimento é necessário para o sucesso no ataque, além da vulnerabilidade de Path Confusion, que é a classe de vulnerabilidade utilizada para a execução do ataque apresentado.

Dito isto, nota-se que o Web Cache Deception Attack é, como o próprio nome sugere, uma classe de ataque e não a vulnerabilidade propriamente dita, sendo essa a Path Confusion.

Serviços de Cache em Web

Serviços de cache possuem a finalidade de armazenar conteúdo não sensível a fim de reduzir o tráfego entre as aplicações clientes (navegador de internet, aplicação móvel etc) e o servidor da aplicação (mais conhecido como backend). Tais serviços podem ser implementados internamente no próprio navegador; ou fornecidos de maneira remota ao cliente, seja na rede local de uma corporação (exemplo: a partir de um servidor de proxy local), através de serviços de internet, como as Redes de Distribuição de Conteúdo (CDN), ou até mesmo pela própria arquitetura da aplicação (como um Load Balancer implementado em um API gateway). Este artigo irá focar somente no ataque que utiliza como infraestrutura as Redes de Distribuição de Conteúdo, mas vale ressaltar que o mesmo princípio pode se aplicar a quaisquer mecanismos de cache de internet apresentados aqui.

Redes de Distribuição de Conteúdo (CDN)

Uma Rede de Distribuição de Conteúdo (Content Delivery Network - CDN) se resume basicamente em um conjunto de servidores espalhados ao redor do globo que tem o principal objetivo de armazenar conteúdos não sensíveis com o intuito de reduzir o tráfego entre o cliente e o servidor principal da aplicação (backend).

Para se garantir que nunca serão armazenados dados sensíveis, os serviços de CDN utilizam como regra de

armazenamento o tipo de conteúdo (mais precisamente pela extensão) do recurso solicitado, ou seja, ele armazena em seu cache local somente dados que são considerados estáticos (teoricamente não sensíveis), como folhas de estilo (CSS), códigos client-side (como Javascript), imagens, textos etc.

Basicamente, a arquitetura de rede da aplicação garante que toda requisição passe pelo serviço do CDN que, além de outras funcionalidades, garante que só enviará para o backend original requisições em que ele realmente não possa fazer o cache (vulgo "cachear") ou ainda não tenha em seu cache.

Na Figura 1, é possível observar o funcionamento de uma rede CDN onde é feita a requisição de um arquivo de folha de estilo (p.ex.: <https://www.empresatld/static/style.css>).

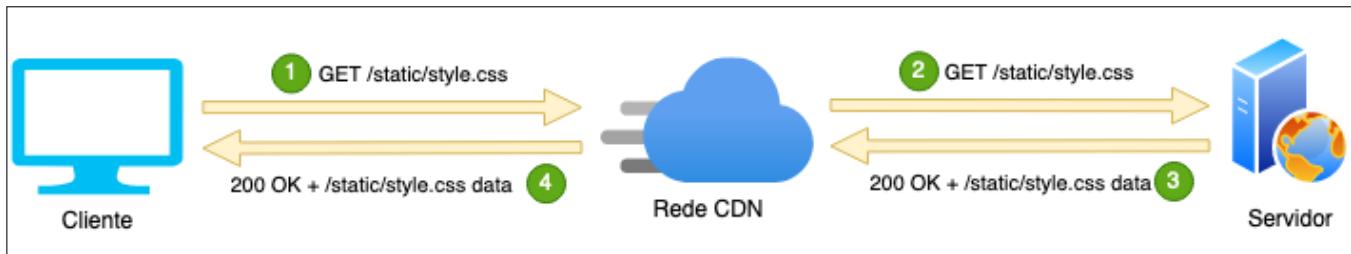


Figura 1: Fluxo de requisição/resposta em CDN sem conteúdo em cache

O diagrama da Figura 1 pode ser descrito nos seguintes passos:

1. A aplicação cliente faz uma requisição a um recursos do servidor (/static/style.css), e essa requisição passa por um servidor do CDN.
2. Caso o conteúdo do recurso não seja "cacheável" ou não exista no cache, o servidor do CDN encaminha a requisição para o servidor original da aplicação (backend). No caso, o conteúdo é "cacheável" por se tratar de uma folha de estilo (*.css).
3. O servidor da aplicação processa a requisição e retorna os dados do recurso solicitado ao CDN. Caso a requisição possua dados estáticos não sensíveis (folhas de estilo, códigos javascript, HTML puro, imagens etc), como é o caso, o servidor do CDN armazena em seu cache local para futuras requisições.
4. Após obter a resposta do conteúdo solicitado, o servidor de CDN envia para a aplicação cliente os dados do recurso solicitado.

Quando qualquer cliente faz uma requisição a um conteúdo estático que já foi visto anteriormente pela rede CDN, tal rede retorna o conteúdo de seu próprio cache local, reduzindo assim o tráfego e o processamento do servidor de aplicação, como pode ser visto na Figura 2.

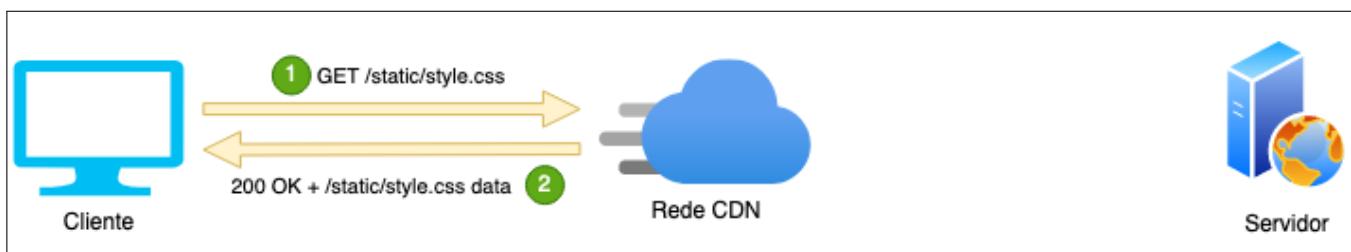


Figura 2: Fluxo de requisição/resposta em CDN com conteúdo em cache

O diagrama da Figura 2 pode ser descrito da seguinte forma:

1. A aplicação cliente faz a solicitação a um conteúdo estático (no caso, um arquivo de estilo em /static/style.css), que passa pela rede CDN.
2. Como a rede CDN já visitou este mesmo conteúdo estático recentemente e o armazenou em seu cache, ela retorna o /static/style.css diretamente do cache, sem a necessidade de solicitar tal recurso ao servidor original da aplicação.

Path Confusion

A vulnerabilidade de confusão de caminho (Path Confusion, em inglês) é uma classe de vulnerabilidade que existe quando a aplicação trata de maneira não adequada, ou de maneira ambígua, um caminho fornecido.

No simples trecho de código da Figura 3, é possível analisar um padrão de URL com expressão regular, onde chama-se a view secret quando o caminho requisitado pelo usuário começar com /api/secret/, como pode ser observado na Figura 4.

```

1 from django.conf.urls import url
2 from api import secret
3
4 urlpatterns = [
5     url(r'^api/secret/$', secret, name='secret')
6 ]

```

Figura 3: Exemplo de código vulnerável a Path Confusion

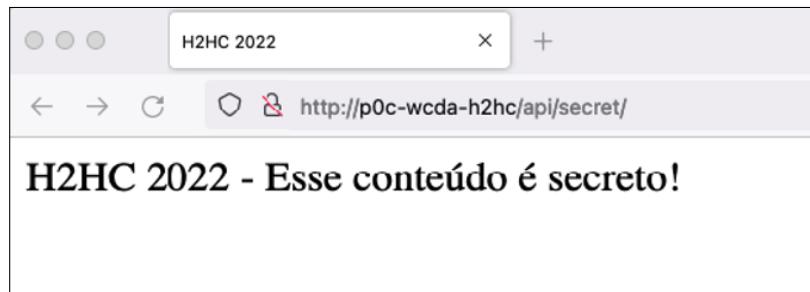


Figura 4: Acesso normal a API em /api/secret/

Entretanto, nota-se que na implementação apresentada, assume-se que a rota deve começar por /api/secret/, mas não há restrição de término da string pela expressão regular (que seria feito pelo caractere \$). Dessa forma, qualquer padrão de URL cujo alvo comece com /api/secret/ dará match na expressão regular, e logo retornará o mesmo conteúdo, de maneira ambígua.

A Figura 5 apresenta o resultado da execução da aplicação caso fosse solicitada a rota em /api/secret/h2hc.css.

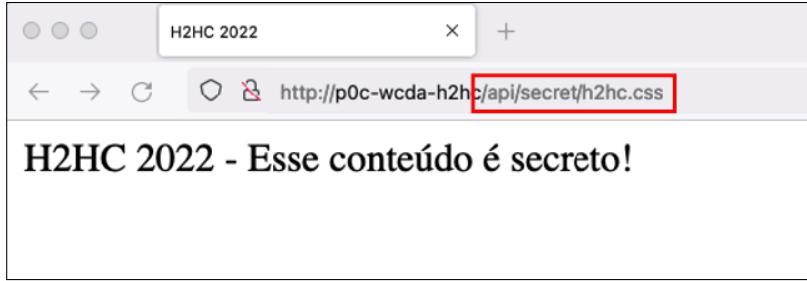


Figura 5: Acesso a API em /api/secret/h2hc.css com resultado ambíguo a /api/secret/

Apesar da forma apresentada ser a mais clássica para exploração dessa classe de vulnerabilidade de Path Confusion, algumas outras variantes também podem ser aplicadas, como por exemplo, os seguintes payloads podem ser utilizados:

- /api/secret/h2hc.css
- /api/secret%0Ah2hc.css (análogo a /api/secret\nh2hc.css)
- /api/secret%3Bh2hc.css (análogo a /api/secret;h2hc.css)
- /api/secret%23h2hc.css (análogo a /api/secret#h2hc.css)
- /api/secret%3Fh2hc.css (análogo a /api/secret?h2hc.css)

Qualquer uma das rotas acima que retornarem o conteúdo de /api/secret/ comprovaria a existência de uma vulnerabilidade Path Confusion.

Web Cache Deception

O ataque de Web Cache Deception foi apresentado por Omer Gil em 2017 [1] [2], e tem como princípio a exploração de uma vulnerabilidade de Path Confusion valendo-se das características de serviços de caching.

De maneira geral, o diagrama da Figura 6 apresenta um fluxo completo de como pode ser executado esse tipo de ataque.

A sequência para o sucesso do ataque em questão pode ser descrita nos seguintes passos:

1. O ataque inicia-se quando o atacante induz um (ou vários) usuário(s) legítimo(s) da aplicação a enviarem alguma requisição para a aplicação afetada. Isso pode ser feito de diversas maneiras distintas, desde engenharia social, infectando algum outro site, ou até mesmo a partir de um ataque de man-in-the-middle na mesma infraestrutura, injetando-se um <iframe> malicioso em qualquer site (em geral, HTTP sem TLS) que os usuários legítimos acessem.
2. Quando o usuário afetado tiver o payload malicioso executado, será feita uma requisição para o backend da aplicação original. Neste momento, assume-se que a aplicação possui algum mecanismo de caching, como CDN, proxy reverso etc. No caso do diagrama acima, será feito uma requisição para a rota /api/secret/h2hc.css e, caso o usuário esteja autenticado e a aplicação utilize controle de sessão por cookies, será garantido a autenticação e autorização.

3. A rede CDN desconhece o caminho /api/secret/h2hc.css, então ela encaminha a requisição para o servidor/backend original, que irá processar e fornecer o resultado.
4. Caso a aplicação esteja vulnerável a Path Confusion (de maneira semelhante à descrita anteriormente), ela interpretará a requisição /api/secret/h2hc.css de forma ambígua a rota /api/secret/, retornando seu resultado para o servidor CDN.
5. Em posse do resultado requisitado pelo cliente, o servidor do CDN retorna para a aplicação cliente os dados solicitados. Como aparentemente a requisição possui um arquivo estático (um arquivo de folha de estilo h2hc.css), a CDN também irá armazenar a resposta do servidor da aplicação para utilizações futuras. Nesse ponto vale notar que a resposta fornecida pelo backend não representa um arquivo estático, mas sim o resultado da chamada à API /api/secret/.
6. Conhecendo o caminho ao qual a vítima foi induzida (em /api/secret/h2hc.css), o atacante então faz a mesma requisição, mas dessa vez sem conter qualquer mecanismo de autenticação ou autorização. A requisição então também passará pela rede CDN.
7. Como o servidor da rede CDN já acessou o recurso anteriormente, será retornado o conteúdo do cache, sem a necessidade de consultar o backend. Entretanto, como a consulta original armazenou no cache da CDN o conteúdo de /api/secret/ devido à vulnerabilidade de Path Confusion, é possível que o atacante obtenha acesso a tal conteúdo, sem a necessidade de estar autenticado na aplicação.

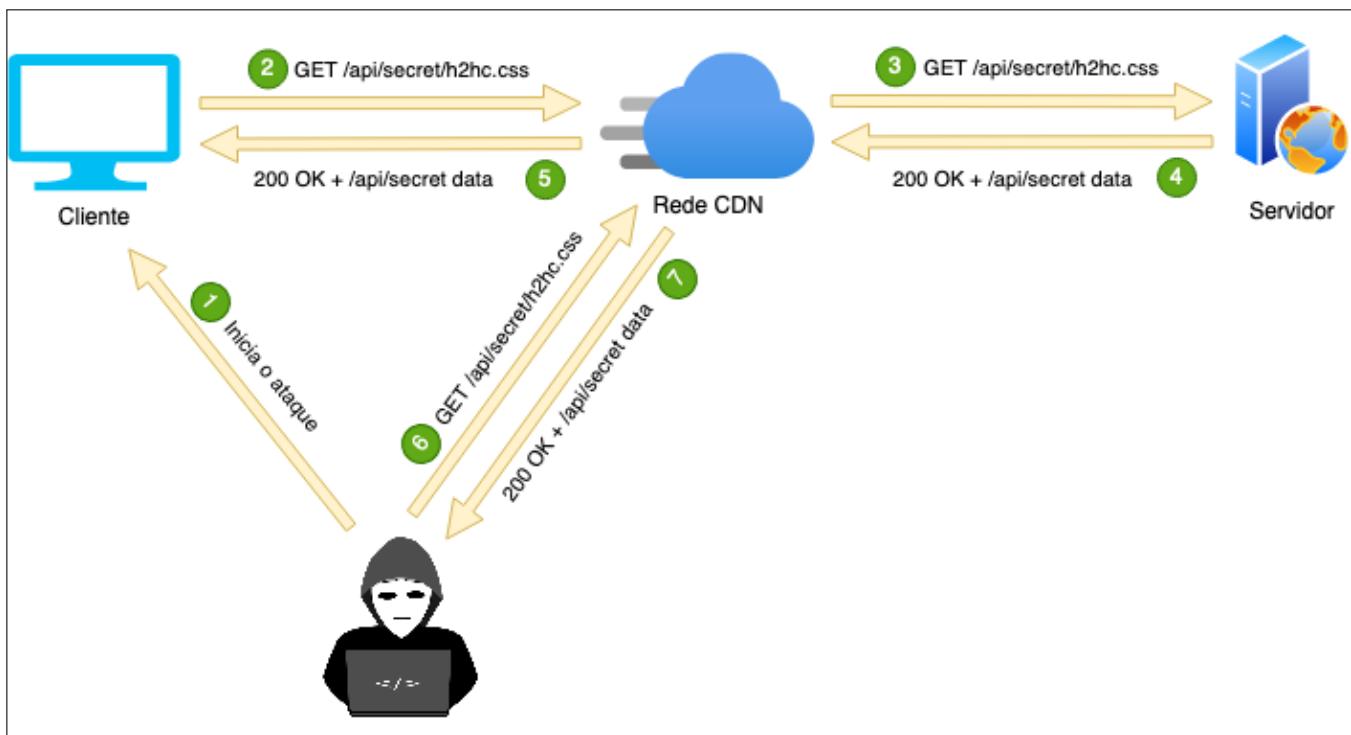


Figura 6: Diagrama do fluxo do ataque de Web Cache Deception

Recomendações

Existem algumas maneiras de se tentar remediar o ataque de Web Cache Deception diretamente no serviço de CDN. Entretanto, ao invés de deixar **somente** um serviço terceiro como responsável pela mitigação da exploração de uma determinada falha (análogo ao uso exclusivo de Web Application Firewalls), a melhor forma de se evitar esse tipo de ataque é através do desenvolvimento seguro, com a mitigação de vulnerabilidades de Path Confusion.

No trecho de código apresentado na Figura 3, a forma mais adequada de se corrigir a vulnerabilidade é em garantir que tanto o início quanto o término do caminho sejam definidos. E isso é feito com a simples modificação da Figura 7, que adiciona o indicador de término (\$) à expressão regular.

```

1 from django.conf.urls import url
2 from api import secrets
3
4 urlpatterns = [
5     url(r'^api/secret/$', secrets, name='secrets') # Fixed
6 ]

```

Figura 7: Correção do código vulnerável a Path Confusion

Nota-se que a simples adição do terminador (\$) da rota faz com que qualquer parâmetro a partir da última barra não dê match com a expressão regular, mitigando a vulnerabilidade e inibindo o ataque apresentado.

Ataque Web Cache Deception in the Wild

O objetivo desse texto não é apresentar de maneira pontual sites vulneráveis a Path Confusion e sujeitos a Web Cache Deception Attack. Porém, um artigo publicado na USENIX Security em 2020 [3] fez um levantamento de sites vulneráveis dentro do Alexa Top Sites [4], que consiste numa lista de sites muito populares. Dentre os Top 5K do Alexa, um total de 295 sites possuem serviços de CDN, sendo 16 desses sites (5,4%) vulneráveis a ataques de Web Cache Deception. É possível analisar este resultado no gráfico da Figura 8.

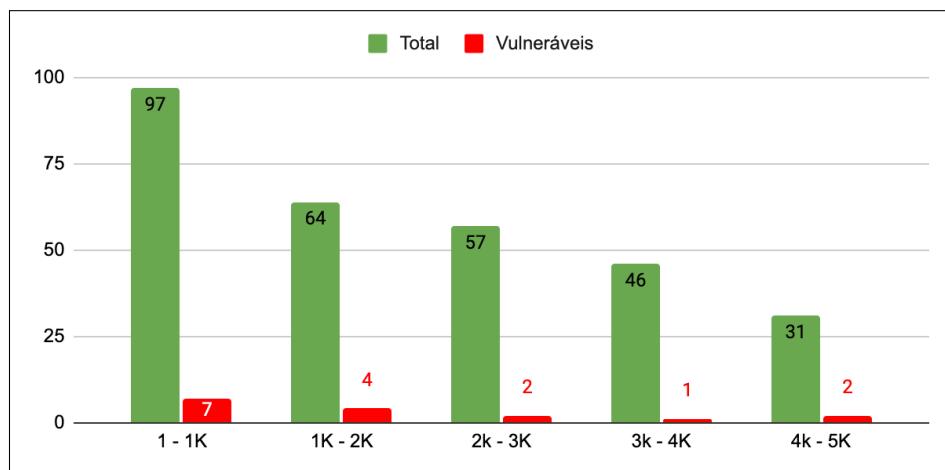


Figura 8: Distribuição de vulnerabilidades no Alexa Top 5K

Mais resultados da pesquisa citada podem ser observados no artigo original.

Conclusão

Esse artigo teve por finalidade apresentar de maneira simples os fundamentos e métodos de exploração de Path Confusion, mais precisamente através do ataque chamado Web Cache Deception. Foram apresentadas as características necessárias para o sucesso no ataque, uma recomendação para mitigar e, apesar de não comum, dados que mostram que essa classe de ataque existe na internet.

Referências

- [1] O. Gil, “Web Cache Deception Attack - Blog,” acessado em 01-Outubro-2022. [Online]. Disponível em: <https://omergil.blogspot.com/2017/02/web-cache-deception-attack.html>
- [2] O. Gil, “Web Cache Deception Attack - Blackhat,” acessado em 01-Outubro-2022. [Online]. Disponível em: <https://www.blackhat.com/docs/us-17/wednesday/us-17-Gil-Web-Cache-Deception-Attack.pdf>
- [3] K. O. Seyed Ali Mirheidari, Sajjad Arshad, “Cached and Confused: Web Cache Deception in the Wild,” acessado em 01-Outubro-2022. [Online]. Disponível em: https://sajjadium.github.io/files/usenixsec2020wcd_paper.pdf
- [4] “Alexa Top 1M,” acessado em 01-Outubro-2022. [Online]. Disponível em: <http://s3.amazonaws.com/alexa-static/top-1m.csv.zip>

Maycon Vitali (OutOfBound) - maycon@hacknroll.com

Maycon Vitali é fundador da Hack N' Roll e atua como Managing Consultant (MC) na PRIDE Security. Com mais de 15 anos de experiência, seus principais interesses incluem segurança ofensiva de aplicações (web, mobile e desktop), testes de invasão (de redes corporativas a ambientes críticos como ICS/SCADA), hacking de embarcados e qualquer coisa relacionada a engenharia reversa. Em seu tempo livre, Maycon dedica-se basicamente em estudar, comer, dormir e apreciar um bom café (sem açúcar, por favor).



Registro Único de Artigo

<https://doi.org/10.47986/16/3>

Introdução

O debugger x64dbg [1] é amplamente utilizado para análise malware. Durante a execução de alguns experimentos com técnicas de anti-debugging, um bug foi identificado no x64dbg que, se explorado, pode se tornar uma técnica de anti-debugging específica para esse debugger.

Esse artigo visa discutir em detalhes o bug encontrado e como o mesmo pode ser explorado para se tornar uma técnica de anti-debugging.

Ambiente utilizado

Para esse artigo, o seguinte ambiente foi utilizado:

- Arquitetura da CPU: amd64
- Sistema operacional: Windows 10 Pro versão 21H2 19044.2006 rodando em uma máquina virtual KVM/Linux.
- Compilador: ml64 do Microsoft Visual Studio Build Tools 2022 17.3.4 [2], onde somente o workload "Desktop development with C++" foi instalado.
- Debugger: x64dbg, snapshot_2022-09-18_20-00.zip

O bug

O código presente na Listagem 1 (teste_bug_1.asm), que simplesmente altera o valor do registrador de segmento FS, será usado para demonstrar o bug.

Para compilar o arquivo teste_bug_1.asm, basta executar no prompt do Visual Studio (x64 Native Tools Command Prompt for VS 2022) o comando comentado na primeira linha do código. Após compilar e carregar o binário no x64dbg, ao se fazer single-step, observa-se a tela da Figura 1 que mostra os seguintes valores de registradores antes da atualização do FS:

- FS = 0x53
- SS = R15W = 0x2B

```
; ml64 teste_bug_1.asm /link /entry:main
```

.code

main proc

nop

nop

nop

; fs = ss

mov r15w, ss

mov fs, r15w

nop

nop

nop

main endp

end

Listagem 1: teste_bug_1.asm

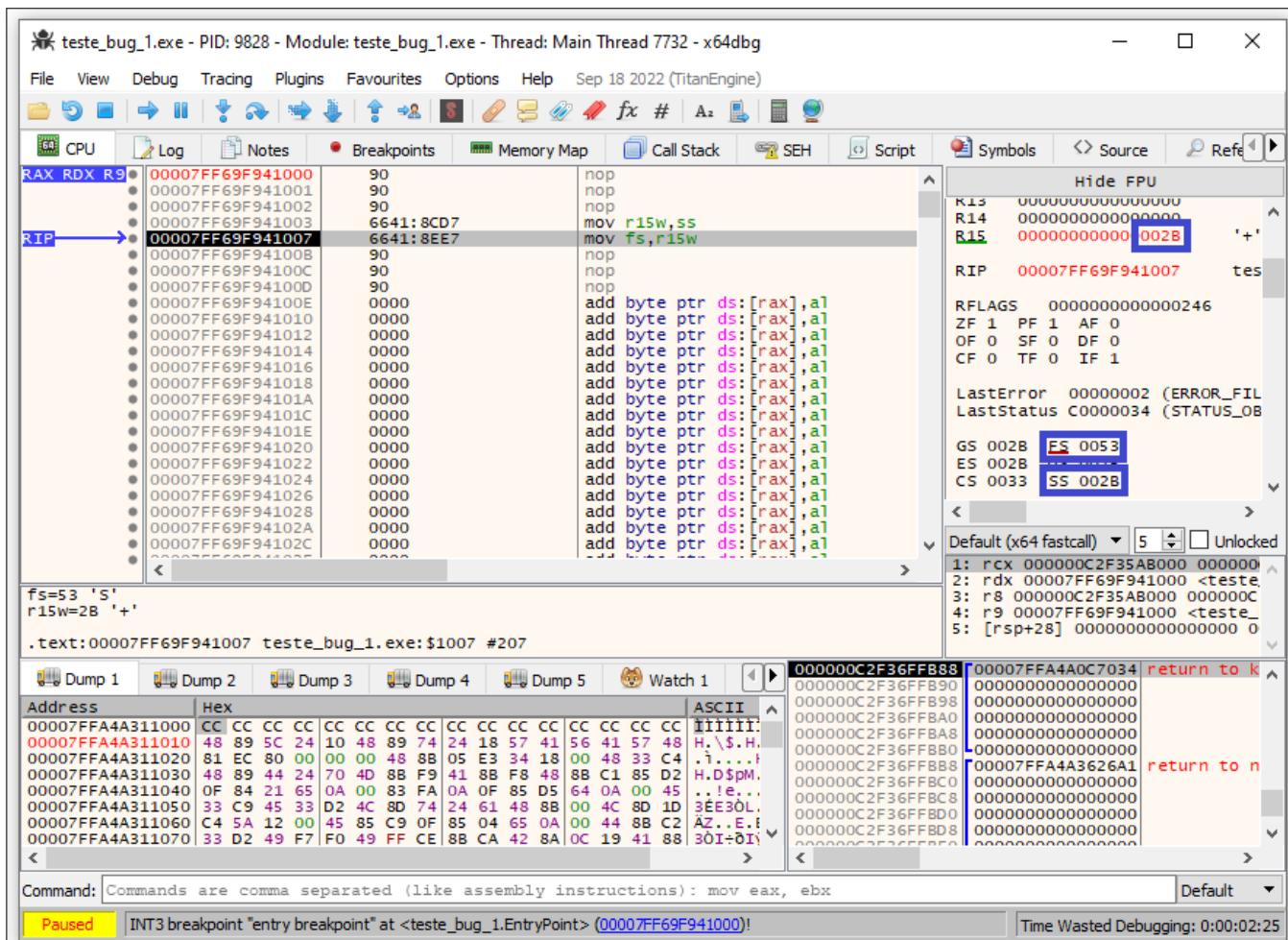


Figura 1: FS, SS e R15W antes do FS ser carregado

A instrução atual mostrada na Figura 1, após executada, deve mudar o FS de forma que FS = SS = R15W = 0x2B. Vamos então fazer um single-step para confirmar que o FS será alterado conforme esperado. A Figura 2 mostra o resultado.

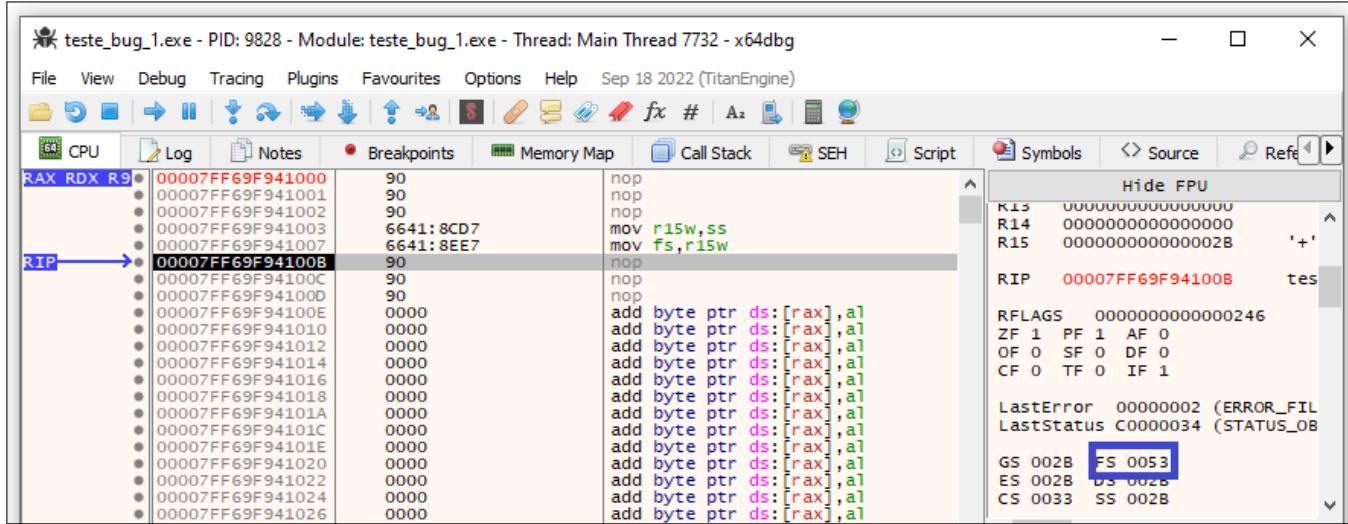


Figura 2: FS, SS e R15W após o FS ser carregado

A técnica de detecção do x64dbg

Como mostra a Figura 2, percebe-se que o valor de FS não foi alterado, o que parece ser um bug no debugger. O que aconteceu? Será que o FS foi de fato alterado mas a interface gráfica não foi atualizada apropriadamente, ou será que o FS não foi sequer alterado? O código da Listagem 2 (teste_bug_2.asm) visa resolver esse problema adicionando uma verificação de forma que um MessageBox será emitido para caso o FS não tenha sido de fato alterado.

Vamos então carregar o binário teste_bug_2.exe no x64dbg e fazer single-step na função main até o FS ser carregado, conforme mostra a Figura 3.

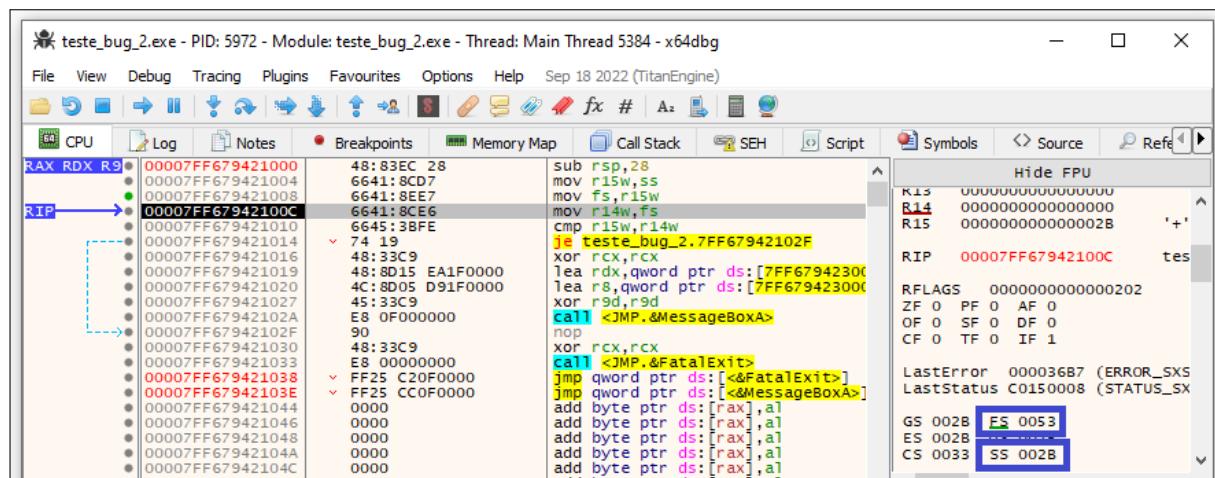


Figura 3: FS, SS e R15W após o FS ser carregado

```

; ml64 teste_bug_2.asm /link /defaultlib:kernel32.lib /defaultlib:user32.lib /entry:main

extern MessageBoxA: proc
extern ExitProcess: proc

.data
mbCaption db "Resultado",0
mbText db "FS nao foi alterado!",0

.code

main proc
    sub rsp, 28h ; calling convention

    ; fs = ss
    mov r15w, ss
    mov fs, r15w

    ; fs == ss ?
    mov r14w, fs
    cmp r15w, r14w
    je igual

diferente:
    xor rcx, rcx ; hWnd
    lea rdx, mbText ; lpText
    lea r8, mbCaption ; lpCaption
    xor r9d, r9d ; uType
    call MessageBoxA

igual:
    nop

    xor rcx, rcx
    call ExitProcess
main endp
end

```

Listagem 2: teste_bug_2.asm

Como esperado, temos o mesmo resultado de antes. Basta então resumir a execução para que o FS seja verificado. O resultado pode ser visto na Figura 4.

Ou seja, está confirmado que não foi somente um erro da interface gráfica do x64dbg pois o FS não foi de fato alterado. Se o mesmo arquivo teste_bug_2.exe for executado fora do x64dbg, ou se for executado dentro do x64dbg mas sem fazer single-step (ou sem pararmos a execução com breakpoint) no carregamento o FS, nenhum MessageBox será exibido.

Temos então um bug no x64dbg que nos permite identificar se carregamentos de FS foram single-stepped ou alvo de um breakpoint. Em outras palavras, temos uma potencial técnica de anti-debugging para um comportamento específico do x64dbg. Fica de desafio aos leitores que testem essa mesma técnica em

outros registradores de segmento, e também que experimentem com single-step e breakpoints em outras instruções do binário (incluindo as instruções adjacentes ao carregamento do FS).

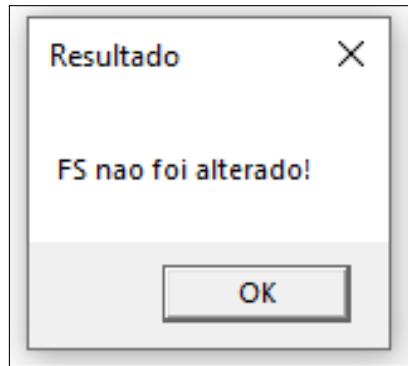


Figura 4: Resultado do teste

Referências

- [1] x64dbg Core Team, “x64dbg,” acessado em 22-Setembro-2022. [Online]. Disponível em: <https://x64dbg.com/>
- [2] Microsoft, “Microsoft Visual Studio Build Tools 2022,” acessado em 22-Setembro-2022. [Online]. Disponível em: <https://visualstudio.microsoft.com/downloads/?q=build+tools>

P3nDr1v3

P3nDr1v3 é um analista de malware com 2 anos de experiência. Antes disso atuou por 1,5 anos como analista de suporte técnico a redes de computadores.

A photograph of a red and white USB flash drive lying on a black computer keyboard. The keyboard keys visible include T, Y, U, G, H, O, L, V, B, M, and a key with a camera icon.

Engenharia Reversa de Software

Autor: Fernando Mercês

Reconstrução da IAT: Quando cortar thunks e excluir nodes?

Registro Único de Artigo

<https://doi.org/10.47986/16/4>

Desabafo

Um motorista da Uber me contou do seu sonho em entrar para a área de TI recentemente. Ele disse que tentou fazer faculdade Ciência da Computação, mas não entendia a matemática e por isso mudou para Análise de Sistemas. Embora eu compreenda, essa sensação de “isso não é pra mim” me preocupa muito e busco dar o meu melhor para provar que computação é para todos.

Não acho que as pessoas devam ter medo de bits, códigos ou apps. Afinal, o computador é uma máquina que faz (na maioria dos casos) somente o que lhe é instruído. É necessário compreendê-lo e você tem o direito de entender exatamente como ele funciona.

Para mim, ser hacker é não se contentar com achismos, sei lás ou explicações levianas. É testar e verificar, empiricamente, comportamentos e só então buscar formas de subvertê-los. No final, ser hacker é sobre não aceitar.

Dito isso, considero de suma importância trabalhos como o da H2HC, que passada época catastrófica que vivenciamos no planeta, está de volta trazendo palestras, treinamentos e networking de alta qualidade para o Brasil de forma acessível e justa. É símbolo de uma resistência que diz “sim, nós podemos”. Meus sinceros agradecimentos a todos os envolvidos neste trabalho tão importante em épocas tão sombrias da educação nacional.

Introdução

Em edições anteriores, falei sobre o unpacking de binários comprimidos por packers - os compressores de executáveis. Na edição número 15 [1], mostrei como vencer a técnica anti-engenharia reversa de redirecionamento de IAT (Import Address Table). Antes de cobrir cenários mais avançados, vou explorar um pouco mais a reconstrução da IAT, a fim de te dar a oportunidade de fundamentar bases mais sólidas que serão muito úteis em desafios mais complexos de unpacking.

Quando comecei a estudar engenharia reversa por volta dos anos 2000, lia muitos tutoriais de unpacking que mostravam como reconstruir a IAT. O problema é que na minha vez de testar o que aprendera, era muito comum que algo não funcionasse por estar diferente.

Para começar, a IAT naturalmente mudava de acordo com o alvo e de acordo com o packer. Além disso, os programas que buscavam a IAT a ser reconstruída também eram diferentes: vários tutoriais mostravam o ImpRec (Imports Reconstructor) [2] [3], que tinha várias versões e opções utilizadas diferentemente. Em 2008, surgiu o CHimpREC (The Cheap Imports Reconstructor) [4] [5], o primeiro público a funcionar com executáveis PE de 64-bits, mas este não teve uma grande aderência da comunidade, até porque o número de executáveis de 64-bits ainda não era tão grande.

A variedade de ferramentas era ótima, mas também representava uma dificuldade para iniciantes que, como eu, não sabiam decidir **qual** ferramenta utilizar e com **quais** opções.

Muitos dos tutoriais disponíveis na época mostravam como remover entradas inválidas da IAT, mas nenhum explicava exatamente quais entradas eu deveria considerar inválidas, nem me davam a base para fazer tal julgamento. Eu mesmo ignorei essa explicação na edição 14 da revista [6], mas agora é hora de me redimir com você e explicar exatamente como essas decisões são tomadas no processo de reconstrução da IAT, que é vital para o unpacking.

Cenário

Para estudar o cenário proposto, você pode fazer download do binário Unpackme2 [7] e encontrar o Original EntryPoint (OEP) dele, como descrito no artigo da edição 13 [8].

Após encontrar o OEP, você sabe que precisa dumper o binário para o disco e reconstruir sua IAT, mas o Scylla [9] - o reconstrutor de IAT mais comum atualmente - acusa entradas inválidas ou suspeitas, como a Figura 1 sugere.

Para saber exatamente o que fazer é necessário investigar a IAT encontrada, mas antes vamos dedicar um tempo na Figura 1 para entender as ricas informações que o Scylla exibe.

Calculando distâncias

A IAT encontrada neste alvo, exibida na Figura 1, começa com seis funções da kernel32.dll seguida de uma função da user32.dll. Depois, há seis funções inválidas e mais duas válidas da kernel32.dll.

Preste atenção nas distâncias entre os nodes: o primeiro node encontrado está no RVA (Relative Virtual Address) 0x2000 e contém seis funções. Como cada ponteiro para função no PE32 tem quatro bytes, faz sentido o próximo node começar no RVA 0x201C (considere um ponteiro nulo no final do array).

NOTA: A natureza l33t das ferramentas de engenharia reversa contribui para que não haja um padrão de nomes seguido por todas elas. No contexto da IAT, o Scylla chama a DLL à qual uma função pertence de **node**, node tree ou FThunk, e uma entrada de função importada de **thunk**. Já o ImpcRec exibe um nome de DLLs como FThunk mas o chama de thunk na opção de excluí-lo. As entradas na IAT para as funções importadas também podem ser chamadas de thunks. Admito que é confuso, mas com as bases apresentadas aqui, você não terá dificuldade de saber do que se trata caso use uma ferramenta diferente.

O terceiro node inicia no RVA 0x2024, o que também faz sentido já que o node anterior tem uma só função

(novamente, considere o elemento nulo no fim do array).

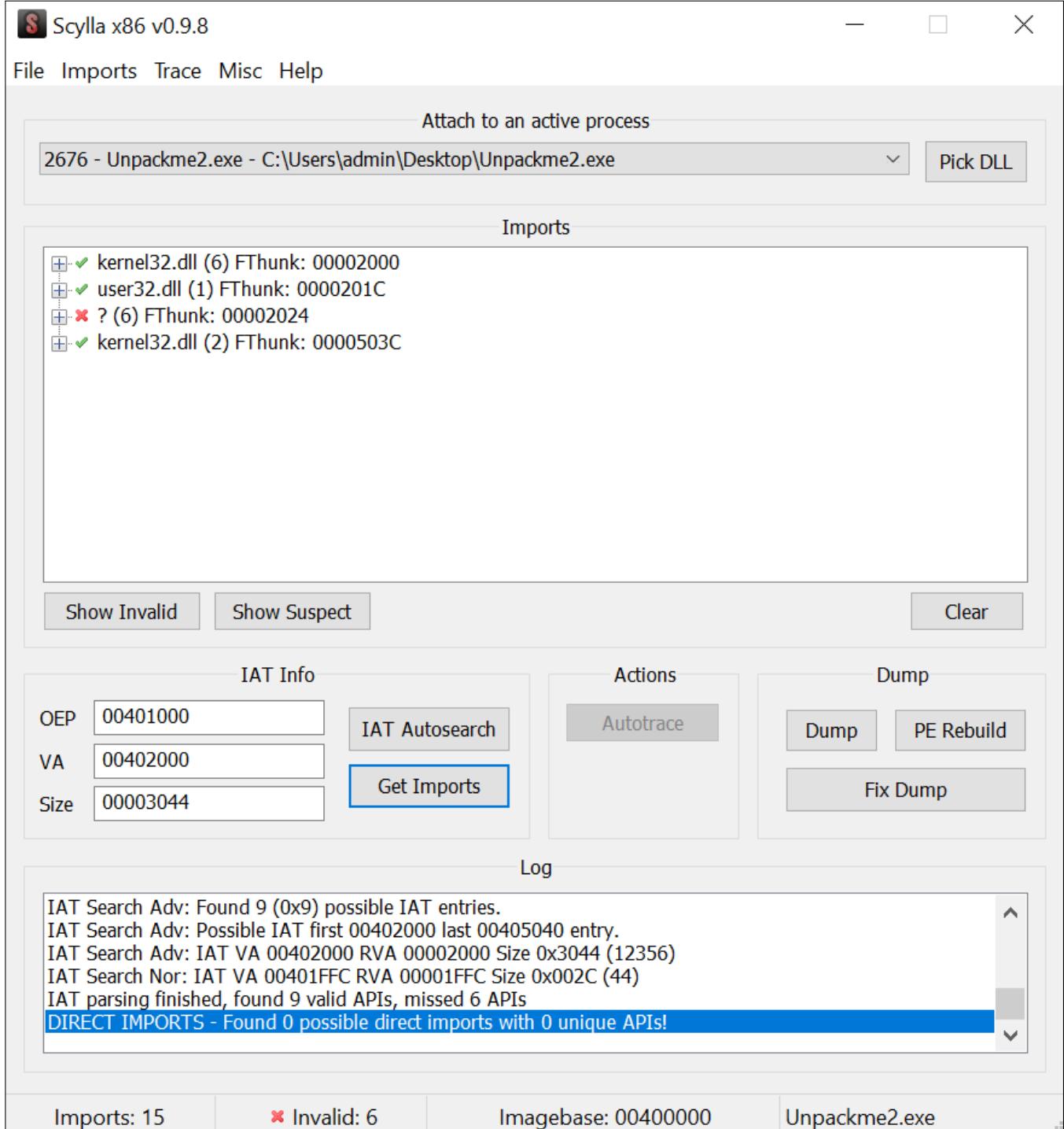


Figura 1: Imports inválidos acusados pelo Scylla

Como o terceiro node possui alegadamente seis funções, para calcular seu tamanho basta fazer 7 (considere o ponteiro nulo no fim do array) vezes 4, que resulta em 0x1C.

Somando-se 0x1C ao RVA 0x2024, concluímos que o próximo node deveria começar no RVA 0x2040, mas o Scylla mostra que o RVA dele é 0x503C - uma distância e tanto! Por aí já se deve desconfiar que o tamanho

da IAT está incorreto também.

Se você dumper o executável e reconstruir a IAT assim, vai ter um executável que não funciona, como pode observar na mensagem de erro da Figura 2 ao tentar executá-lo.

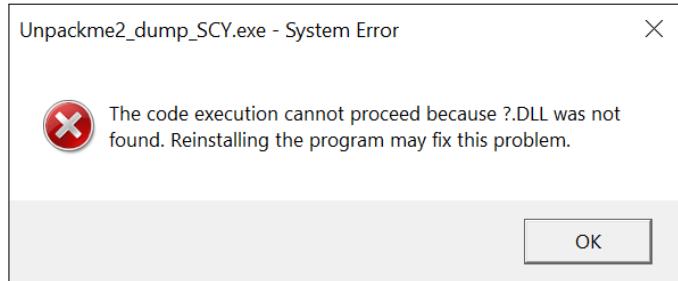


Figura 2: Erro ao tentar executar o binário dumpado e fixado com a IAT proposta pelo Scylla

Torna-se necessário então manipular a IAT encontrada, utilizando-se para tal os recursos do Scylla. No entanto, para saber exatamente o que fazer, é necessário investigar.

Investigando a IAT

Para investigar a IAT, copie o VA (Virtual Address) dela e observe-o no dump do x64dbg [10]. Para facilitar, clique com o botão direito no dump e escolha a exibição **Address**. O dump resolverá na coluna **Comments** o nome da função apontada, caso haja.

A Figura 3 mostra as funções encontradas em cada node, numeradas de 1 a 3. Note que o fim do terceiro node não aparece na imagem.

Address	Value	ASCII	Comments
00402000	75110B60	..u	kernel32.GetLocalTime
00402004	75109910	...u	kernel32.TerminateProcess
00402008	75112E80	...u	kernel32.GetCurrentProcess
0040200C	75111720	..u	kernel32.SetUnhandledExceptionFilter
00402010	75124F20	O.u	kernel32.UnhandledExceptionFilter
00402014	75110B70	p..u	kernel32.IsProcessorFeaturePresent
00402018	00000000	
0040201C	76790EC0	A.yv	user32.MessageBoxW
00402020	00000000	
00402024	0040116E	n.@.	
00402028	00000000	
0040202C	00000000	
00402030	00403008	.0@.	

Figura 3: Nodes da IAT com seus thunks (ponteiros para funções)

Perceba que os endereços dos thunks do terceiro node não fazem sentido. O primeiro aponta para 0x40116E, que é um endereço na faixa do próprio executável e não de uma função externa a ela. O segundo

está zerado. Isso significa que sim, você pode excluir o node inteiro com a opção **Delete tree node** como mostra a Figura 4.

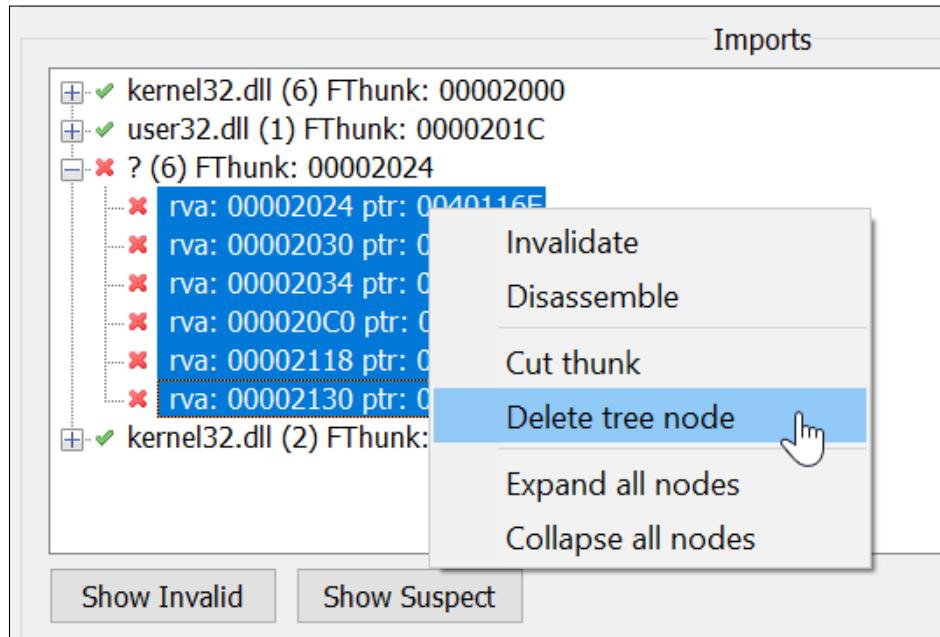


Figura 4: Deletando o terceiro node

Agora vamos para o quarto e último node, que tá bem estranho por conta da sua distância dos outros. O RVA dele é 0x503c, portanto o VA é 0x40503c (RVA + ImageBase). A Figura 5 mostra seu conteúdo no dump.

0040503C	75110A60	...u	kernel32.GetModuleHandleA
00405040	7510F550	Põ.u	kernel32GetProcAddress
00405044	00000000	
00405048	76790EC0	À.yv	user32.MessageBoxW
0040504C	00000000	
00405050	00000000	
00405054	00000000	

Figura 5: Conteúdo do quarto node

Apesar de conter endereços válidos, ele não é um node que faz sentido, pois numa IAT normal, os nodes são contíguos, separados apenas por um elemento nulo. Como tem vários bytes de distância entre o terceiro node e este, você pode deletá-lo com segurança, mesmo que o Scylla ache que ele é válido.

No fim, sua IAT deve ficar como ilustrado pela Figura 6.

Agora sim você pode dumper e reconstruir a IAT com o botão **Fix Dump** do Scylla. O executável gerado deve funcionar sem problemas como mostra a Figura 7.

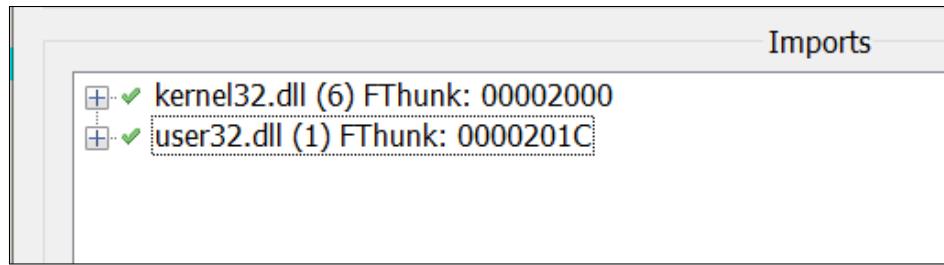


Figura 6: IAT correta, depois de ter o terceiro e o quarto nodes excluídos

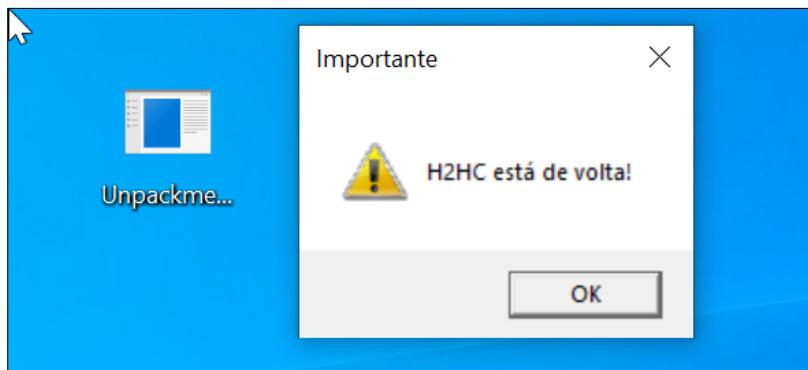


Figura 7: Execução com sucesso do binário descomprimido e com IAT corrigida

Conclusão

Recomendo que você treine o que foi apresentado aqui, afinal, sabemos que a leitura com o tempo vai embora da memória. Você também pode construir seus próprios programas e comprimí-los com packers como o MPRESS [11] [12], UPX [13] ou FSG [14] [15], todos muito parecidos. Até a próxima!

Referências

- [1] H. Magazine, “H2HC Magazine 15a Edição,” acessado em 01-Outubro-2022. [Online]. Disponível em: <https://github.com/h2hconference/H2HCMagazine/tree/master/15>
- [2] MackT/uCF2000, “ImpRec (Imports Reconstructor),” acesso não testado. [Online]. Disponível em: <http://forum.exetools.com/showthread.php?t=11454>
- [3] MackT/uCF2000, “ImpRec (Imports Reconstructor),” acessado em 01-Outubro-2022. [Online]. Disponível em: <https://www.mentebinaria.com.br/files/file/63-imprec/>
- [4] S. Doucet, “CHimpREC (The Cheap Imports Reconstructor,” acesso não testado. [Online]. Disponível em: <https://recon.cx/2008/speakers.html#i64bitunpacking>
- [5] S. Doucet, “CHimpREC (The Cheap Imports Reconstructor,” acessado em 01-Outubro-2022. [Online]. Disponível em: <https://www.mentebinaria.com.br/files/file/2-chimprec/>

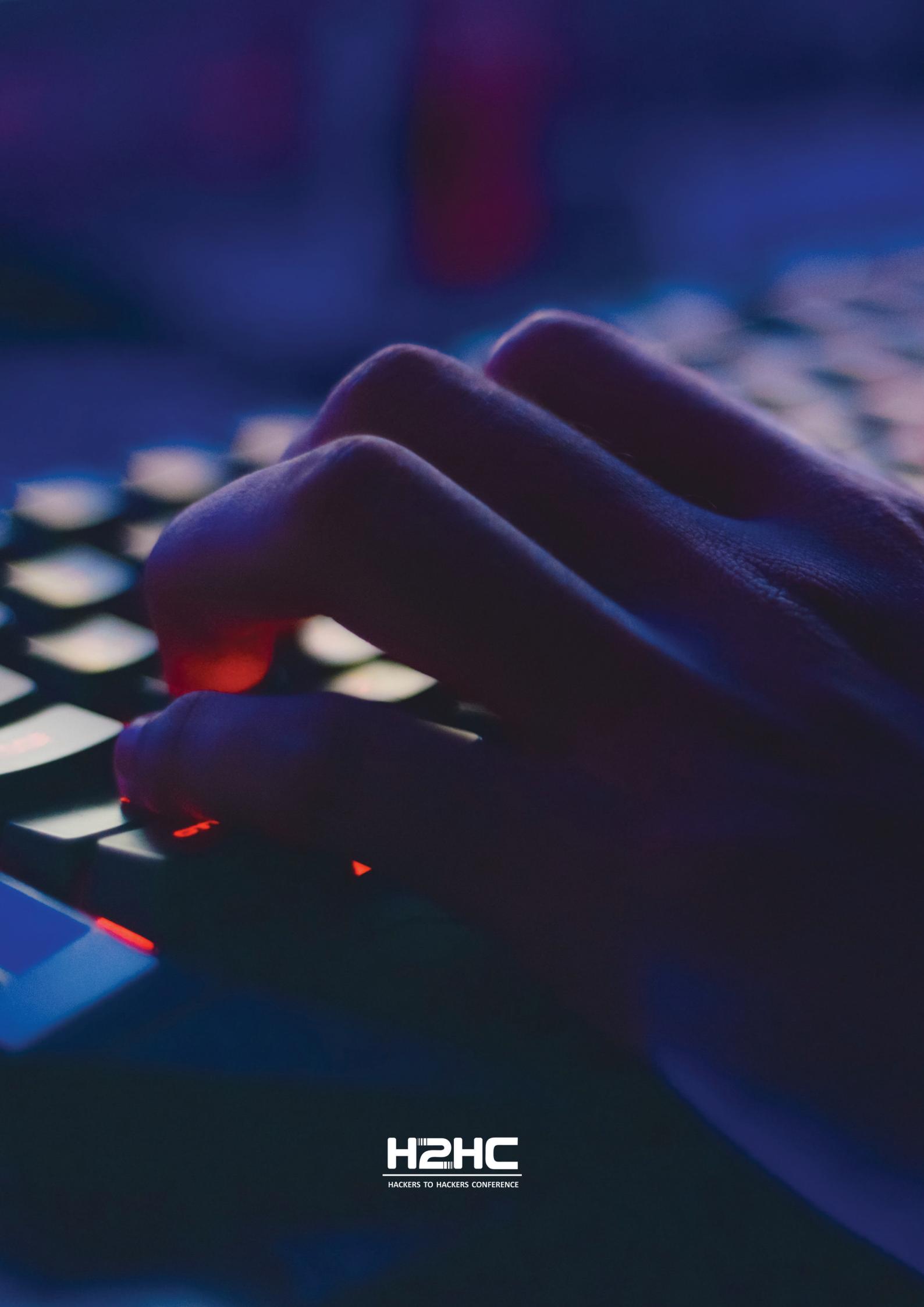
- [6] H. Magazine, "H2HC Magazine 14a Edição," acessado em 01-Outubro-2022. [Online]. Disponível em: https://www.h2hc.com.br/revista/RevistaH2HC_14.pdf
- [7] F. Mercês, "Unpackme2," acessado em 05-Outubro-2022. [Online]. Disponível em: <https://github.com/h2hconference/H2HCMagazine/tree/master/16>
- [8] H. Magazine, "H2HC Magazine 13a Edição," acessado em 01-Outubro-2022. [Online]. Disponível em: https://www.h2hc.com.br/revista/RevistaH2HC_13.pdf
- [9] S. C. Team, "Scylla," acessado em 01-Outubro-2022. [Online]. Disponível em: <https://github.com/NtQuery/Scylla>
- [10] x64dbg Core Team, "x64dbg," acessado em 01-Outubro-2022. [Online]. Disponível em: <https://x64dbg.com/>
- [11] MATCODE, "MPRESS," acessado em 01-Outubro-2022. [Online]. Disponível em: <https://www.matcode.com/mpress.htm>
- [12] MATCODE, "MPRESS," acessado em 01-Outubro-2022. [Online]. Disponível em: <https://www.mentebinaria.com.br/files/file/64-mpress/>
- [13] L. M. . J. F. R. Markus F.X.J. Oberhummer, "UPX," acessado em 01-Outubro-2022. [Online]. Disponível em: <https://upx.github.io>
- [14] Xtreeme, "FSG," acesso não testado. [Online]. Disponível em: <http://www.xtreeme.prv.pl>
- [15] Xtreeme, "FSG," acessado em 01-Outubro-2022. [Online]. Disponível em: <https://www.mentebinaria.com.br/files/file/24-fsg/>

Fernando Mercês

Fernando é Pesquisador de Ameaças na Trend Micro, onde atua como investigador de ciber crime, utilizando engenharia reversa e técnicas de inteligência de ameaças no time de Pesquisa de Ameaças Futuras (FTR). Criador de várias ferramentas livres na área, com frequência apresenta suas pesquisas nos principais eventos de segurança no Brasil e no exterior. É também professor e fundador da Mente Binária, uma instituição de ensino e pesquisa sem fins lucrativos comprometida com o ensino de computação no Brasil.







H2HC

HACKERS TO HACKERS CONFERENCE