Seminarthemen WS 2017/2018

am Lehrstuhl für Netz- und Datensicherheit

10. Oktober 2017

1 Zeitplan

Vorbesprechung:	10.10.2017 um 14:15 Uhr
Bewerbung mit einem Exposee:	24.10.2017
Acceptance Notification	27.10.2017
Abgabe Preversion	18.12.2017
Abgabe finale Version	29.01.2018
Präsentationen	Blockveranstaltung nach Doodle-Umfrage
Meldung der Ergebnisse an das Prüfungsamt	Anfang des folgenden Semesters

2 Angebotene Themen

PDF Mirage: Content Masking Attack Against Information-Based Online Services [1]

Art: **Bachelor/Master**
Betreuer: Marcus Niemietz

We present a new class of content masking attacks against the Adobe PDF standard, causing documents to appear to humans dissimilar to the underlying content extracted by information-based services. We show three attack variants with notable impact on real-world systems. Our first attack allows academic paper writers and reviewers to collude via subverting the automatic reviewer assignment systems in current use by academic conferences including INFOCOM, which we reproduced. Our second attack renders ineffective plagiarism detection software, particularly Turnitin, targeting specific small plagiarism similarity scores to appear natural and evade detection. In our final attack, we place masked content into the indexes for Bing, Yahoo!, and DuckDuckGo which renders as information entirely different from the keywords used to locate it, enabling spam, profane, or possibly illegal content to go unnoticed by these search engines but still returned in unrelated search results. Lastly, as these systems eschew optical character recognition (OCR) for its overhead, we offer a comprehensive and lightweight alternative mitigation method.

Link: `https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/markwood`

Loophole: Timing Attacks on Shared Event Loops in Chrome [2]

Art: **Bachelor/Master**
Betreuer: Marcus Niemietz

Event-driven programming (EDP) is the prevalent paradigm for graphical user interfaces, web clients, and it is rapidly gaining importance for server-side and network programming. Central components of EDP are event loops, which act as FIFO queues that are used by processes to store and dispatch messages received from other processes.

In this paper we demonstrate that shared event loops are vulnerable to side-channel attacks, where a spy process monitors the loop usage pattern of other processes by enqueueing events and measuring the time it takes for them to be dispatched. Specifically, we exhibit attacks against the two central event loops in Google's Chrome web browser: that of the I/O thread of the host process, which multiplexes all network events and user actions, and that of the main thread of the renderer processes, which handles rendering and Javascript tasks.

For each of these loops, we show how the usage pattern can be monitored with high resolution and low overhead, and how this can be abused for malicious purposes, such as web page identification, user behavior detection, and covert communication.

Stacco: Differentially Analyzing Side-Channel Traces for Detecting SSL/TLS Vulnerabilities in Secure Enclaves [3]

Art:	**Master**
Betreuer:	Robert Merget

Intel Software Guard Extension (SGX) offers software applications a shielded execution environment, dubbed enclave, to protect their confidentiality and integrity from malicious operating systems. As processors with this extended feature become commercially available, many new software applications are developed to enrich to the SGX-enabled software ecosystem. One important primitive for these applications is a secure communication channel between the enclave and a remote trusted party. The SSL/TLS protocol, which is the de facto standard for protecting transport-layer network communications, has been broadly regarded a natural choice for such purposes. However, in this paper, we show that the marriage between SGX and SSL may not be a smooth sailing. Particularly, we consider a category of side-channel attacks against SSL/TLS implementations in secure enclaves, which we call the control-flow inference attacks. In these attacks, the malicious operating system kernel may perform a powerful man-in-the-kernel attack to collect execution traces of the enclave programs at the page level, the cacheline level, or the branch level, while positioning itself in the middle of the two communicating parties. At the center of our work is a differential analysis framework, dubbed Stacco, to dynamically analyze the SSL/TLS implementations and detect vulnerabilities—discernible execution traces—that can be exploited as decryption oracles. Surprisingly, in spite of the prevailing constant-time programming paradigm adopted by many cryptographic libraries, we found exploitable vulnerabilities in the latest versions of all the SSL/TLS libraries we have examined. To validate the detected vulnerabilities, we developed a man-in-the-kernel adversary to demonstrate Bleichenbacher attacks against the latest OpenSSL library running in the SGX enclave (with the help of Graphene) and completely broke the PreMasterSecret encrypted by a 4096-bit RSA public key with only 57,286 queries. We also conducted CBC padding oracle attacks against the latest GnuTLS running in Graphene-SGX and an open-source SGX-implementation of mbedTLS (i.e., mbedTLS-SGX) that runs directly inside the enclave, and showed that it only needs 48,388 and 25,717 queries, respectively, to break one block of AES ciphertext. Empirical evaluation suggests these man-in-the-kernel attacks can be completed within one or two hours.

Directed Greybox Fuzzing [4]

Art: **Bachelor**
Betreuer: Robert Merget

Existing Greybox Fuzzers (GF) cannot be effectively directed, for instance, towards problematic changes or patches, towards critical system calls or dangerous locations, or towards functions in the stacktrace of a reported vulnerability that we wish to reproduce. In this paper, we introduce Directed Greybox Fuzzing (DGF) which generates inputs with the objective of reaching a given set of target program locations efficiently. We develop and evaluate a simulated annealing-based power schedule that gradually assigns more energy to seeds that are closer to the target locations while reducing energy for seeds that are further away. Experiments with our implementation AFLGo demonstrate that DGF outperforms both directed symbolic-execution-based whitebox fuzzing and undirected greybox fuzzing. We show applications of DGF to patch testing and crash reproduction, and discuss the integration of AFLGo into Google's continuous fuzzing platform OSS-Fuzz. Due to its directedness, AFLGo could find 39 bugs in several well-fuzzed, security-critical projects like LibXML2. 17 CVEs were assigned.

Global Measurement of DNS Manipulation [5]

Art: **Bachelor/Master**
Betreuer: Jens Müller

Despite the pervasive nature of Internet censorship and the continuous evolution of how and where censorship is applied, measurements of censorship remain comparatively sparse. Understanding the scope, scale, and evolution of Internet censorship requires global measurements, performed at regular intervals. Unfortunately, the state of the art relies on techniques that, by and large, require users to directly participate in gathering these measurements, drastically limiting their coverage and inhibiting regular data collection. To facilitate large-scale measurements that can fill this gap in understanding, we develop Iris, a scalable, accurate, and ethical method to measure global manipulation of DNS resolutions. Iris reveals widespread DNS manipulation of many domain

SmartPool: Practical Decentralized Pooled Mining [6]

Art:	**Bachelor/Master**
Betreuer:	Martin Grothe

Cryptocurrencies such as Bitcoin and Ethereum are operated by a handful of mining pools. Nearly 95% of Bitcoin's and 80% of Ethereum's mining power resides with less than ten and six mining pools respectively. Although miners benefit from low payout variance in pooled mining, centralized mining pools require members to trust that pool operators will remunerate them fairly. Furthermore, centralized pools pose the risk of transaction censorship from pool operators, and open up possibilities for collusion between pools for perpetrating severe attacks.

In this work, we propose SMARTPOOL, a novel protocol design for a decentralized mining pool. Our protocol shows how one can leverage smart contracts, autonomous blockchain programs, to decentralize cryptocurrency mining. SMARTPOOL gives transaction selection control back to miners while yielding low-variance payouts. SMARTPOOL incurs mining fees lower than centralized mining pools and is designed to scale to a large number of miners. We implemented and deployed a robust SMARTPOOL implementation on the Ethereum and Ethereum Classic networks. To date, our deployed pools have handled a peak hashrate of 30 GHs from Ethereum miners, resulting in 105 blocks, costing miners a mere 0:6% of block rewards in transaction fees.

A Longitudinal, End-to-End View of the DNSSEC Ecosystem [7]

Art:	**Bachelor/Master**
Betreuer:	Martin Grothe

The Domain Name System's Security Extensions (DNSSEC) allow clients and resolvers to verify that DNS responses have not been forged or modified inflight. DNSSEC uses a public key infrastructure (PKI) to achieve this integrity, without which users can be subject to a wide range of attacks. However, DNSSEC can operate only if each of the principals in its PKI properly performs its management tasks: authoritative name servers must generate and publish their keys and signatures correctly, child zones that support DNSSEC must be correctly signed with their parent's keys, and resolvers must actually validate the chain of signatures.

This paper performs the first large-scale, longitudinal measurement study into how well DNSSEC's PKI is managed. We use data from all DNSSEC-enabled subdomains under the .com, .org, and .net TLDs over a period of 21 months to analyze DNSSEC deployment and management by domains; we supplement this with active measurements of more than 59K DNS resolvers worldwide to evaluate resolver-side validation.

Our investigation reveals pervasive mismanagement of the DNSSEC infrastructure. For example, we found that 31% of domains that support DNSSEC fail to publish all relevant records required for validation; 39% of the domains use insufficiently strong key-signing keys; and although 82% of resolvers in our study request DNSSEC records, only 12% of them actually attempt to validate them. These results highlight systemic problems, which motivate improved automation and auditing of DNSSEC management.

The first collision for full SHA-1 [8]

Art: **Bachelor/Master**
Betreuer: Matthias Horst

Abstract. SHA-1 is a widely used 1995 NIST cryptographic hash function standard that was officially deprecated by NIST in 2011 due to fundamental security weaknesses demonstrated in various analyses and theoretical attacks. Despite its deprecation, SHA-1 remains widely used in 2017 for document and TLS certificate signatures, and also in many software such as the GIT versioning system for integrity and backup purposes.

We were able to find this collision by combining many special cryptanalytic techniques in complex ways and improving upon previous work. In total the computational effort spent is equivalent to 263.1 calls to SHA-1's compression function, and took approximately 6 500 CPU years and 100 GPU years. While the computational power spent on this collision is larger than other public cryptanalytic computations, it is still more than 100 000 times faster than a brute force search.

Cryptography with Updates [9]

Art: **Master**
Betreuer: Matthias Horst

Starting with the work of Bellare, Goldreich and Goldwasser [CRYPTO'94], a rich line of work has studied the design of updatable cryptographic primitives. For example, in an updatable signature scheme, it is possible to efficiently transform a signature over a message into a signature over a related message without recomputing a fresh signature. In this work, we continue this line of research, and perform a systematic study of updatable cryptography. We take a unified approach towards adding updatability features to recently studied cryptographic objects such as attribute-based encryption, functional encryption, witness encryption, indistinguishability obfuscation, and many others that support non-interactive computation over inputs. We, in fact, go further and extend our approach to classical protocols such as zero-knowledge proofs and secure multiparty computation. To accomplish this goal, we introduce a new notion of updatable randomized encodings that extends the standard notion of randomized encodings to incorporate updatability features. We show that updatable randomized encodings can be used to generically transform cryptographic primitives to their updatable counterparts. We provide various definitions and constructions of updatable randomized encodings based on varying assumptions, ranging from one-way functions to compact functional encryption.

Practical Keystroke Timing Attacks in Sandboxed JavaScript [10]

Art: **Bachelor/Master**
Betreuer: Christopher Späth

Keystrokes trigger interrupts which can be detected through software side channels to reconstruct keystroke timings. Keystroke timing attacks use these side channels to infer typed words, passphrases, or create user fingerprints. While keystroke timing attacks are considered harmful, they typically require native code execution to exploit the side channels and, thus, may not be practical in many scenarios.

In this paper, we present the first generic keystroke timing attack in sandboxed JavaScript, targeting arbitrary other tabs, processes and programs. This violates same-origin policy, HTTPS security model, and process isolation. Our attack is based on the interrupt-timing side channel which has previously only been exploited using native code. In contrast to previous attacks, we do not require the victim to run a malicious binary or interact with the malicious website. Instead, our attack runs in a background tab, possibly in a minimized browser window, displaying a malicious online advertisement. We show that we can observe the exact inter-keystroke timings for a user's PIN or password, infer URLs entered by the user, and distinguish different users time-sharing a computer. Our attack works on personal computers, laptops and smartphones, with different operating systems and browsers. As a solution against all known JavaScript timing attacks, we propose a fine-grained permission model.

Towards Evaluating the Robustness of Neural Networks [11]

Art: **Bachelor/Master**
Betreuer: Juraj Somorovsky

Neural networks provide state-of-the-art results for most machine learning tasks. Unfortunately, neural networks are vulnerable to adversarial examples: given an input x and any target classification t, it is possible to find a new input x' that is similar to x but classified as t. This makes it difficult to apply neural networks in security-critical areas. Defensive distillation is a recently proposed approach that can take an arbitrary neural network, and increase its robustness, reducing the success rate of current attacks' ability to find adversarial examples from 95% to 0.5%.

In this paper, we demonstrate that defensive distillation does not significantly increase the robustness of neural networks by introducing three new attack algorithms that are successful on both distilled and undistilled neural networks with 100% probability. Our attacks are

tailored to three distance metrics used previously in the literature, and when compared to previous adversarial example generation algorithms, our attacks are often much more effective (and never worse). Furthermore, we propose using high-confidence adversarial examples in a simple transferability test we show can also be used to break defensive distillation. We hope our attacks will be used as a benchmark in future defense attempts to create neural networks that resist adversarial examples.

unCaptcha: A Low-Resource Defeat of reCaptcha's Audio Challenge [12]

Art:	**Bachelor**
Betreuer:	Dennis Felsch

CAPTCHAs are the Internet's first line of defense against automated account creation and service abuse. Google's reCaptcha, one of the most popular captcha systems, is currently used by hundreds of thousands of websites to protect against automated attackers by testing whether a user is truly human. This paper presents unCaptcha, an automated system that can solve reCaptcha's most difficult auditory challenges with high success rate. We evaluate unCaptcha using over 450 reCaptcha challenges from live websites, and show that it can solve them with 85.15% accuracy in 5.42 seconds, on average. unCaptcha combines free, public, online speech-to-text engines with a novel phonetic mapping technique, demonstrating that it requires minimal resources to mount a large-scale successful attack on the reCaptcha system.

The Password Reset MitM Attacks [13]

Art:	**Bachelor/Master**
Betreuer:	Vladislav Mladenov

We present the password reset MitM (PRMitM) attack and show how it can be used to take over user accounts. The PRMitM attack exploits the similarity of the registration and password reset processes to launch a man in the middle (MitM) attack at the application level. The attacker initiates a password reset process with a website and forwards every challenge to the victim who either wishes to register in the attacking site or to access a particular resource on it. The attack has several variants, including exploitation of a password reset process that relies on the victim's mobile phone, using either SMS or phone call. We evaluated the PRMitM attacks on Google and Facebook users in several experiments, and found that their password reset process is vulnerable to the PRMitM attack. Other websites and some popular mobile applications are vulnerable as well. Although solutions seem trivial in some cases, our experiments show that the straightforward solutions are not as effective as expected. We designed and evaluated two secure password reset processes and evaluated them on users of Google and Facebook. Our results indicate a significant improvement in the security. Since millions of accounts are currently vulnerable to the PRMitM attack, we also present a list of recommendations for implementing and auditing the password reset process.

Link: `https://www.ieee-security.org/TC/SP2017/papers/207.pdf`

Literatur

[1] I. Markwood, D. Shen, Y. Liu, and Z. Lu, "PDF mirage: Content masking attack against information-based online services," in *26th USENIX Security Symposium (USENIX Security 17)*, (Vancouver, BC), pp. 833–847, USENIX Association, 2017.

[2] P. Vila and B. Kopf, "Loophole: Timing attacks on shared event loops in chrome," in *26th USENIX Security Symposium (USENIX Security 17)*, (Vancouver, BC), pp. 849–864, USENIX Association, 2017.

[3] Y. Xiao, M. Li, S. Chen, and Y. Zhang, "Stacco: Differentially analyzing side-channel traces for detecting SSL/TLS vulnerabilities in secure enclaves," *CoRR*, vol. abs/1707.03473, 2017.

[4] M. Böhme, V.-T. Pham, M.-D. Nguyen, and A. Roychoudhury, "Directed greybox fuzzing," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS'17)*, 2017.

[5] P. Pearce, B. Jones, F. Li, R. Ensafi, N. Feamster, N. Weaver, and V. Paxson, "Global measurement of dns manipulation," in *26th {USENIX} Security Symposium ({USENIX} Security 17)*, USENIX Association, 2017.

[6] L. Luu, Y. Velner, J. Teutsch, and P. Saxena, "Smartpool: Practical decentralized pooled mining," in *26th USENIX Security Symposium (USENIX Security 17)*, (Vancouver, BC), pp. 1409–1426, USENIX Association, 2017.

[7] T. Chung, R. van Rijswijk-Deij, B. Chandrasekaran, D. Choffnes, D. Levin, B. M. Maggs, A. Mislove, and C. Wilson, "A longitudinal, end-to-end view of the DNSSEC ecosystem," in *26th USENIX Security Symposium (USENIX Security 17)*, (Vancouver, BC), pp. 1307–1322, USENIX Association, 2017.

[8] P. Ananth, A. Cohen, and A. Jain, "Cryptography with updates." Cryptology ePrint Archive, Report 2016/934, 2016. `http://eprint.iacr.org/2016/934`.

[9] M. Stevens, E. Bursztein, P. Karpman, A. Albertini, and Y. Markov, "The first collision for full sha-1," in *Crypto*, IACR, 2017.

[10] M. Lipp, D. Gruss, M. Schwarz, D. Bidner, C. Maurice, and S. Mangard, "Practical keystroke timing attacks in sandboxed javascript," in *European Symposium on Research in Computer Security*, pp. 191–209, Springer, 2017.

[11] N. Carlini and D. A. Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*, pp. 39–57, 2017.

[12] K. Bock, D. Patel, G. Hughey, and D. Levin, "uncaptcha: A low-resource defeat of recaptcha's audio challenge," in *11th USENIX Workshop on Offensive Technologies (WOOT 17)*, (Vancouver, BC), USENIX Association, 2017.

[13] N. Gelernter, S. Kalma, B. Magnezi, and H. Porcilan, "The password reset mitm attack," in *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*, pp. 251–267, 2017.