Seminarthemen SS 2017

am Lehrstuhl für Netz- und Datensicherheit

5. April 2017

1 Zeitplan

Vorbesprechung:	18.04.2017 um 15:00 Uhr
Bewerbung mit einem Exposee:	02.05.2017
Acceptance Notification	08.05.2017
Abgabe Preversion	19.06.2017
Abgabe finale Version	28.07.2017
Präsentationen	Blockveranstaltung nach Doodle-Umfrage
Meldung der Ergebnisse an das Prüfungsamt	Anfang des folgenden Semesters

2 Angebotene Themen

SoK: XML Parser Vulnerabilities [1]

Art:	**Bachelor**
Betreuer:	Christopher Späth

The Extensible Markup Language (XML) has become a widely used data structure for web services, Single-Sign On, and various desktop applications. The core of the entire XML processing is the XML parser. Attacks on XML parsers, such as the Billion Laughs and the XML External Entity (XXE) Attack are known since 2002. Nevertheless even experienced companies such as Google, and Facebook were recently affected by such vulnerabilities. In this paper we systematically analyze known attacks on XML parsers and deal with challenges and solutions of them. Moreover, as a result of our in-depth analysis we found three novel attacks. We conducted a large-scale analysis of 30 different XML parsers of six different programming languages. We created an evaluation framework that applies different variants of 17 XML parser attacks and executed a total of 1459 attack vectors to provide a valuable insight into a parser's configuration. We found vulnerabilities in 66 % of the default configuration of all tested

parses. In addition, we comprehensively inspected parser features to prevent the attacks, show their unexpected side effects, and propose secure configurations.

Verifiable Message-Locked Encryption[2]

Art: **Master**
Betreuer: Sebastian Lauer

One of today's main challenge related to cloud storage is to maintain the functionalities and the efficiency of customers' and service providers' usual environments, while protecting the confidentiality of sensitive data. Deduplication is one of those functionalities: it enables cloud storage providers to save a lot of memory by storing only once a file uploaded several times. But classical encryption blocks deduplication. One needs to use a "message-locked encryption" (MLE), which allows the detection of duplicates and the storage of only one encrypted file on the server, which can be decrypted by any owner of the file. However, in most existing scheme, a user can bypass this deduplication protocol. In this article, we provide servers verifiability for MLE schemes: the servers can verify that the ciphertexts are well-formed. This property that we formally define forces a customer to prove that she complied to the deduplication protocol, thus preventing her to deviate from the prescribed functionality of MLE. We call it deduplication consistency. To achieve this deduplication consistency, we provide (i) a generic transformation that applies to any MLE scheme and (ii) an ElGamal-based deduplication-consistent MLE, which is secure in the random oracle model.

Measurement and Analysis of Private Key Sharing in the HTTPS Ecosystem [3]

Art: **Bachelor**
Betreuer: Vladislav Mladenov

The semantics of online authentication in the web are rather straightforward: if Alice has a certificate binding Bob's name to a public key, and if a remote entity can prove knowledge of Bob's private key, then (barring key compromise) that remote entity must be Bob. However, in reality, many websites—and the majority of the most popular ones-are hosted at least in part by third parties such as Content Delivery Networks (CDNs) or web hosting providers. Put simply: administrators of websites who deal with (extremely) sensitive user data are giving their private keys to third parties. Importantly, this sharing of keys is undetectable by most users, and widely unknown even among researchers. In this paper, we perform a

large-scale measurement study of key sharing in today's web. We analyze the prevalence with which websites trust third-party hosting providers with their secret keys, as well as the impact that this trust has on responsible key management practices, such as revocation. Our results reveal that key sharing is extremely common, with a small handful of hosting providers having keys from the majority of the most popular websites. We also find that hosting providers often manage their customers' keys, and that they tend to react more slowly yet more thoroughly to compromised or potentially compromised keys.

NEZHA: Efficient Domain-Independent Differential Testing[4]

Art: **Master**
Betreuer: Robert Merget

Differential testing uses similar programs as cross-referencing oracles to find semantic bugs that do not exhibit explicit erroneous behaviors like crashes or assertion failures. Unfortunately, existing differential testing tools are domain specific and inefficient, requiring large numbers of test inputs to find a single bug. In this paper, we address these issues by designing and implementing NEZHA, an efficient input-format-agnostic differential testing framework. The key insight behind NEZHA's design is that current tools generate inputs by simply borrowing techniques designed for finding crash or memory corruption bugs in individual programs (e.g., maximizing code coverage). By contrast, NEZHA exploits the behavioral asymmetries between multiple test programs to focus on inputs that are more likely to trigger semantic bugs. We introduce the notion of Delta-diversity, which summarizes the observed asymmetries between the behaviors of multiple test applications. Based on Delta-diversity, we design two efficient domain-independent input generation mechanisms for differential testing, one gray-box and one black-box. We demonstrate that both of these input generation schemes are significantly more efficient than existing tools at finding semantic bugs in real-world, complex software.

An Efficient Approach to Detect TorrentLocker Ransomware in Computer Systems [5]

Art: **Bachelor**
Betreuer: Matthias Horst

TorrentLocker is a ransomware that encrypts sensitive data located on infected computer systems. Its creators aim to ransom the victims, if they want to retrieve their data. Unfortunately, antiviruses have difficulties to detect such polymorphic malware. In this paper, we propose a novel approach to detect online suspicious processes accessing a large number of

files and encrypting them. Such a behavior corresponds to the classical scenario of a malicious ransomware. We show that the Kullback-Liebler divergence can be used to detect with high effectiveness whether a process transforms structured input files (such as JPEG files) into unstructured encrypted files, or not. We focus mainly on JPEG files since irreplaceable pictures represent in many cases the most valuable data on personal computers or smartphones.

Is There an Oblivious RAM Lower Bound? [6]

Art:	**Master**
Betreuer:	Paul Rösler

An Oblivious RAM (ORAM), introduced by Goldreich and Ostrovsky (JACM 1996), is a (probabilistic) RAM that hides its access pattern, i.e. for every input the observed locations accessed are similarly distributed. Great progress has been made in recent years in minimizing the overhead of ORAM constructions, with the goal of obtaining the smallest overhead possible.

We revisit the lower bound on the overhead required to obliviously simulate programs, due to Goldreich and Ostrovsky. While the lower bound is fairly general, including the offline case, when the simulator is given the reads and writes ahead of time, it does assume that the simulator behaves in a "balls and bins" fashion. That is, the simulator must act by shuffling data items around, and is not allowed to have sophisticated encoding of the data.

We prove that for the offline case, showing a lower bound without the above restriction is related to the size of the circuits for sorting. Our proof is constructive, and uses a bit-slicing approach which manipulates the bit representations of data in the simulation. This implies that without obtaining yet unknown superlinear lower bounds on the size of such circuits, we cannot hope to get lower bounds on offline (unrestricted) ORAMs.

A Formal Security Analysis of the Signal Messaging Protocol [7]

Art:	**Master**
Betreuer:	Paul Rösler

Signal is a new security protocol and accompanying app that provides end-to-end encryption for instant messaging. The core protocol has recently been adopted by WhatsApp, Facebook Messenger, and Google Allo among many others; the first two of these have at least 1 billion active users. Signal includes several uncommon security properties (such as "future secrecy"

or "post-compromise security"), enabled by a novel technique called ratcheting in which session keys are updated with every message sent. Despite its importance and novelty, there has been little to no academic analysis of the Signal protocol.

We conduct the first security analysis of Signal's Key Agreement and Double Ratchet as a multi-stage key exchange protocol. We extract from the implementation a formal description of the abstract protocol, and define a security model which can capture the "ratcheting" key update structure. We then prove the security of Signal's core in our model, demonstrating several standard security properties. We have found no major flaws in the design, and hope that our presentation and results can serve as a starting point for other analyses of this widely adopted protocol.

Transcript Collision Attacks: Breaking Authentication in TLS, IKE, and SSH

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

In response to high-profile attacks that exploit hash function collisions, software vendors have started to phase out the use of MD5 and SHA-1 in third-party digital signature applications such as X.509 certificates. However, weak hash constructions continue to be used in various cryptographic constructions within mainstream protocols such as TLS, IKE, and SSH, because practitioners argue that their use in these protocols relies only on second preimage resistance, and hence is unaffected by collisions. This paper systematically investigates and debunks this argument. We identify a new class of transcript collision attacks on key exchange protocols that rely on efficient collision- finding algorithms on the underlying hash constructions. We implement and demonstrate concrete credentialforwarding attacks on TLS 1.2 client authentication, TLS 1.3 server authentication, and TLS channel bindings. We describe almost-practical impersonation and downgrade attacks in TLS 1.1, IKEv2 and SSH-2. As far as we know, these are the first collision-based attacks on the cryptographic constructions used in these popular protocols. Our practical attacks on TLS were responsibly disclosed (under the name SLOTH) and have resulted in security updates to several TLS libraries. Our analysis demonstrates the urgent need for disabling all uses of weak hash functions in mainstream protocols, and our recommendations have been incorporated in the upcoming Token Binding and TLS 1.3 protocols. [8]

TLS in the Wild: An Internet-wide Analysis of TLS-based Protocols for Electronic Communication

Art: **Bachelor**
Betreuer: Martin Grothe

Email and chat still constitute the majority of electronic communication on the Internet. The standardisation and acceptance of protocols such as SMTP, IMAP, POP3, XMPP, and IRC has allowed to deploy servers for email and chat in a decentralised and interoperable fashion. These protocols can be secured by providing encryption with TLS—directly or via the STARTTLS extension. X.509 PKIs and ad hoc methods can be leveraged to authenticate communication peers. However, secure configuration is not straight-forward and many combinations of encryption and authentication mechanisms lead to insecure deployments and potentially compromise of data in transit. In this paper, we present the largest study to date that investigates the security of our email and chat infrastructures. We used active Internet-wide scans to determine the amount of secure service deployments, and employed passive monitoring to investigate to which degree user agents actually choose secure mechanisms for their communication. We addressed both client-to-server interactions as well as server-to-server forwarding. Apart from the authentication and encryption mechanisms that the investigated protocols offer on the transport layer, we also investigated the methods for client authentication in use on the application layer. Our findings shed light on an insofar unexplored area of the Internet. Our results, in a nutshell, are a mix of both positive and negative findings. While large providers offer good security for their users, most of our communication is poorly secured in transit, with weaknesses in the cryptographic setup and especially in the choice of authentication mechanisms. We present a list of actionable changes to improve the situation. [9]

Literatur

[1] Spaeth, "Sok: Xml parser vulnerabilities." https://www.usenix.org/system/files/conference/woot16/woot16-paper-spath.pdf, 2016.

[2] S. Canard, F. Laguillaumie, and M. Paindavoine, *Verifiable Message-Locked Encryption*, pp. 299–315. Cham: Springer International Publishing, 2016.

[3] F. Cangialosi, T. Chung, D. Choffnes, D. Levin, B. M. Maggs, A. Mislove, and C. Wilson, "Measurement and analysis of private key sharing in the https ecosystem," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, (New York, NY, USA), pp. 628–640, ACM, 2016.

[4] S. J. S. A. D. K. Theofilos Petsios, Adrian Tang and S. Jana, "NEZHA: Efficient Domain-independent Differential Testing," in *Proceedings of the 38th IEEE Symposium on Security & Privacy*, (San Jose, CA), May 2017.

[5] F. Mbol, J.-M. Robert, and A. Sadighian, *An Efficient Approach to Detect TorrentLocker Ransomware in Computer Systems*, pp. 532–541. Cham: Springer International Publishing, 2016.

[6] E. Boyle and M. Naor, "Is there an oblivious RAM lower bound?," in *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science, Cambridge, MA, USA, January 14-16, 2016*, pp. 357–368, 2016.

[7] K. Cohn-Gordon, C. Cremers, B. Dowling, L. Garratt, and D. Stebila, "A formal security analysis of the signal messaging protocol," *IACR Cryptology ePrint Archive*, vol. 2016, p. 1013, 2016.

[8] K. Bhargavan and G. Leurent, "Transcript collision attacks: Breaking authentication in tls, ike, and ssh," in *Network and Distributed System Security Symposium–NDSS 2016*, 2016.

[9] R. Holz, J. Amann, O. Mehani, M. Wachs, and M. A. Kaafar, "Tls in the wild: an internet-wide analysis of tls-based protocols for electronic communication," *arXiv preprint arXiv:1511.00341*, 2015.