April 11, 2016

Question 1  \textit{DNSSEC} \hspace{1cm} (10 \text{ min})

In class you learned about DNSSEC which uses certificate-style authentication for DNS results.

(a) In the case of a negative result (the name requested doesn’t exist), what is the result returned by the nameserver to avoid dynamically signing a statement such as “\texttt{aaa.google.com} does not exist”? (This should be a review from lecture.)

\begin{quote}
\textbf{Solution:} The nameserver has a canonical ordering of all record names in its zone. It creates, off-line, signed statements for each pair of adjacent names in the ordering. When a request comes in for which there is no name, the nameserver replies with the record that lists the two existing names just before and just after where the requested name would be in the ordering. This proves the non-existence of the requested name. The reply is called an NSEC record.
\end{quote}

(b) How could an attacker use this feature to enumerate all the record names in a particular zone. Could this be a security concern?

\begin{quote}
\textbf{Solution:} Revealing this information could aid in other attacks. For example, the names in a zone could be used as targets when probing for vulnerable servers.
\end{quote}

(c) How could you change the response sent by the nameserver to mitigate this issue?

\begin{quote}
\textbf{HINT:} One of the crypto primitives you learned about will be helpful.

\textbf{Solution:} The nameserver can create a list of the hash of each name, ordered by hash, and sign pairs of adjacent hash values. When a request comes in, the nameserver will compute the hash of the requested name and return the signed record that lists the two existing hashed names just before and just after where the hash of the requested name would be. The client can then compute the hash of the requested name and see that it would be between the two values returned by the server if it existed. This type of record has been proposed as a replacement to the NSEC record in DNSSEC; it is called an NSEC3 record.

This is not a perfect solution, because an attacker who has a list of candidate hostnames can still perform offline guessing attacks. These attacks might still
allow an attacker to recover some or many of the hostnames in a particular zone.

Question 2  

**DNSSEC / TLS**  

(a) Oski wants to securely communicate with CalBears.com using TLS. Which of the following entities must Oski trust in order to communicate with confidentiality, integrity, and authenticity?

1. The operators of CalBears.com  
2. Oski’s computer  
3. Cryptographic algorithms  
4. Computers on Oski’s local network  
5. The operators of CalBears.com’s authoritative DNS servers  
6. The entire network between Oski and CalBears.com  
7. CalBears.com’s Cert. Auth. (CA)  
8. All of the CA’s that come configured into Oski’s browser  
9. All of the CA’s that come configured into CalBears.com’s software  
10. The operators of .com’s Authoritative DNS servers  
11. The operators of the Authoritative DNS root servers

**Solution:** (1) The operators of CalBears.com, (2) Oski’s computer, (3) Cryptographic algorithms, (7) CalBears.com’s Certificate Authority, (8) All of the CA’s that come configured into Oski’s browser.

(b) Suppose we didn’t want to trust any of the existing CA’s, but DNSSEC was widely deployed and we were willing to trust DNSSEC and the operators of the root zone and of .com. How could TLS be modified, to avoid the need to trust any of the existing CA’s, under these conditions?

**Solution:** The basic idea would be for a TLS client to retrieve a site’s public keys via DNSSEC records from the site’s domain, rather than via a certificate sent by the server and signed by a CA. Such an approach could also instead return signatures of public keys that the server would then still send to the TLS client; the client would now validate the public key based on the signature received via DNSSEC rather than some CA. The inspiration for this question came from DNS-based Authentication of Named Entities (DANE). DANE is a standard currently under development that, among other things, allows certificates to be bound to DNSSEC records.

(c) Assume end-to-end DNSSEC deployment as well as full deployment of your change. Oski wants to securely communicate with CalBears.com using TLS. What changes
are there to the list in part A (eg. what must Oski trust in order to communicate with confidentiality, integrity, and authenticity)?

Solution: No longer need to trust: (8) All of the CA’s that are configured in Oski’s browser, (7) CalBears.com Certificate Authority
Also need to trust: (5) The operators of CalBears.com’s authoritative DNS servers, (10) The operators of .com’s authoritative DNS servers, (11) The operators of the authoritative DNS root servers.

(d) Is this change good or bad? List at least one positive and one negative effect that would result from this change.

Solution: Many answers are possible here. One could say that it’s a good change because there are now fewer parties to trust. Another solution is that it’s a good change because it associates trust directly with parties associated with a domain, rather than with all CA’s. But one could also argue that now the operators of the root name servers gain a great deal of power.

Question 3  Port Scanning and Defenses  (5 min)
Having taken CS161, you’ve become an expert in computer security. You’re hired by Famouc Inc. to test their software for vulnerabilities. Their primary product is special server software. For administrative purposes, this software allows authenticated admin access on a special port p.

You discover that the proprietary third-party library they use for authentication has a bug. As a result, anyone who knows p can gain admin access to the software. Famous Inc. can’t update the third-party library immediately, but they still want to protect the users of their software. So, they thus introduce a configuration option into their software (which they control) that sets p to be a random 16-bit number.

Is this strategy an effective defense? If not, how could an attacker gain admin access?

Solution: An attacker can quickly probe a particular computer to check which ports are “open”. This means that when a port p on a remote computer is probed, the probed computer replies with an answer indicating that a service is listening on a particular port. This is called port scanning, and is in itself not a necessarily malicious activity.

Question 4  Detecting attacks  (7 min)
Suppose that S is a network-based intrusion detector that works by passively analyzing individual UDP and TCP packets. Suppose that A is a host-based intrusion detector that is a component of the browser that processes and analyzes individual URLs before they are loaded by the browser.
Your company decides to build a hybrid scheme for detecting malicious URLs. The hybrid scheme works by combining scheme $S$ and scheme $A$, running both in parallel on the same traffic. The combination could be done in one of two ways. Scheme $H_E$ would generate an alert if for a given network connection either scheme $S$ or scheme $A$ generates an alert. Scheme $H_B$ would generate an alert only if both scheme $S$ and scheme $A$ generate an alert for the same connection. (Assume that there is only one URL in each network connection.)

(a) Assuming that decisions made by $S$ and $A$ are well-modeled as independent processes, and ignoring any concerns regarding evasion, what can you say about the false positives and false negatives of $H_B$ and $H_E$?

**Solution:** The key insight here is that alarms by $H_B$ will be a subset of the alarms generated by $H_E$. Since $H_B$ will generate fewer alarms for non-malicious activities, it will have less false positives. On the other hand, because it generates fewer alarms, it might miss more malicious activity, implying more false negatives.

(b) If deploying the hybrid scheme in a new environment, is one of $H_E$ and $H_B$ clearly better?

**Solution:** In the absence of more data, particularly the cost of false positive and false negatives, as well as the rate of malicious and non-malicious activity, it is impossible to make any decision.