Paxson
Spring 2013

CS 161
Computer Security

Discussion 11

April 10, 2013

Question 1 ***DNSSEC*** **(7 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 "`aaa.google.com` does not exist"? (This should be a review from lecture.)

Solution: The nameserver uses a canonical alphabetical 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** resource record.

For example, suppose the following names exist in `google.com` when it's viewed in alphabetical order:

```
...
a-one-and-a-two-and-a-three-and-a-four.google.com
a1sauce.google.com
aardvark.google.com
...
```

In this ordering, `aaa.google.com` would fall between `a1sauce.google.com` and `aardvark.google.com`. So in response to a DNSSEC query for `aaa.google.com`, the name server would return an NSEC RR that in informal terms states "the name that in alphabetical order comes after `a1sauce.google.com` is `aardvark.google.com`", along with a signature of that NSEC RR made using `google.com`'s key.

The signature allows the recipient to verify the validity of the statement, and by checking that `aaa.google.com` would have fallen between those two names, the recipient has confidence that the name indeed does not exist.

(b) One drawback with this approach is that an attacker can now enumerate all the record names in a zone. Why is this a security concern?

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.

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

HINT: One of the crypto primitives you learned about will be helpful.

Solution: Instead of sorting on the domains, the sorting is done on *hashes* of the names. For example, suppose the procedure is to use SHA1 and then sort the output treated as hexadecimal digits. If the original zone contained:

```
  barkflea.foo.com
   boredom.foo.com
    bug-me.foo.com
   galumph.foo.com
   help-me.foo.com
perplexity.foo.com
     primo.foo.com
```

then the corresponding SHA1 values would be:

```
  barkflea.foo.com = e24f2a7b9fa26e2a0c201a7196325889abf7c45b
   boredom.foo.com = 6d0edfd3efa5bf11b094cb26a7c95a3bd5e85a84
    bug-me.foo.com = 649bb99765bb29c379d935a68db2eebc95ad6a29
   galumph.foo.com = 71d0549ab66459447a62b639849145dace1fa68e
   help-me.foo.com = 1ed14d3733f88e5794cd30cbbef8cc32fa47db2a
perplexity.foo.com = 446ac4777f8d3883da81631902fafd0eba3064ec
     primo.foo.com = 8a1011003ade80461322828f3b55b46c44814d6b
```

Sorting these on the hex for the hashes:

```
   help-me.foo.com = 1ed14d3733f88e5794cd30cbbef8cc32fa47db2a
perplexity.foo.com = 446ac4777f8d3883da81631902fafd0eba3064ec
    bug-me.foo.com = 649bb99765bb29c379d935a68db2eebc95ad6a29
   boredom.foo.com = 6d0edfd3efa5bf11b094cb26a7c95a3bd5e85a84
   galumph.foo.com = 71d0549ab66459447a62b639849145dace1fa68e
     primo.foo.com = 8a1011003ade80461322828f3b55b46c44814d6b
  barkflea.foo.com = e24f2a7b9fa26e2a0c201a7196325889abf7c45b
```

Now if a client requests a lookup of `snup.foo.com`, which doesn't exist, the name server will return a record that in informal terms states "the hash that in alphabetical order comes after `71d0549ab66459447a62b639849145dace1fa68e` is `8a1011003ade80461322828f3b55b46c44814d6b`" (again along with a signature made using `foo.com`'s key). This type of Resource Record is called **NSEC3**.

The client would compute the SHA1 hash of `snup.foo.com`:

`snup.foo.com = 81a8eb88bf3dd1f80c6d21320b3bc989801caae9`

and verify that in alphabateical order it indeed falls between those two returned values (standard ASCII sorting collates digits as coming before letters). That confirms the non-existence of `snup.foo.com` but without indicating what names *do* exist, just what hashes exist.

By using a cryptographically strong hash function like SHA1, it's believed infeasible to reverse the hash function to find out what name(s) appear in the zone (there's more than one potential name because hash functions are many-to-one). Note though that an attacker can still conduct a *dictionary attack*, either directly trying names to see whether they exist, or inspecting the hash values returned by NSEC3 RRs to determine whether names in a dictionary (for which the attacker computes hash values offline) indeed appear in the domain.

Question 2 ***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:** Any 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 fewer 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 know which is better.

Question 3 ***Base Rate Fallacy*** **(7 min)**

The Department of Homesand Security is really concerned about the threat of "sanders," basically people who come to American Shores and leave with sand, a precious national resource. Deeming this unacceptable, the senate authorizes a project to automatically identify sanders crossing into the country, tentatively called "Operation Sanders Crossing" or OSX for short. As part of OSX, you are tasked to choose a Sander Detection System from the many offered by defense contractors.

(a) Let S denote the event that a sander is crossing the border. Let A denote the event that the sander detection system generates an alarm. In what follows, identify which is the term that is the correct notation for false positive, false negative, true positive and true negative.

$$\mathbb{P}[A \mid S]$$
$$\mathbb{P}[\neg A \mid S]$$
$$\mathbb{P}[A \mid \neg S]$$
$$\mathbb{P}[\neg A \mid \neg S]$$

Solution: In order, they are: true positive, false negative, false positive, true negative.

(b) One of the candidate systems advertises a false negative rate of 1 in 10,000 and a false positive rate of 1 in 10,000. Your boss thinks this rate is pretty darn good and thinks they should be bought. What do you think?

Solution: In the absence of knowing the base rate at which actual events occur, it is impossible to make an informed decision.

(c) In terms of the notation above, write down the notation for the *Bayesian detection rate*, i.e., the probability that an alarm is actually for a sander.

Solution: $\mathbb{P}[S \mid A]$

In the IDS scenarios discussed in lecture, this is the probability that there is actually an attack, when your IDS generates an alarm. Note that this is different from the true positive rate, which is the probability of an alarm if an attack takes place. Since attacks are often uncommon, the true positive rate can easily be misinterpreted in terms of what it says about the practicality of a detector.

(d) If a doctor tells you that you tested positive for something and that the test is 99% accurate, does that mean you have a 99% chance of having the disease? What is the chance of having a disease if the disease is really rare, say only 1 in 1 million people have the disease ? You can assume that the false negative and false positive rate for this test are both 1%.

Solution: No, it does not mean you have a 99% chance of having the disease. The chances are more like 1 in 10,000.

Let's say 1 million people take the test. We know that only 1 person has the disease out of the 1 million. How many will test positive? Well, out of the 999,999 people without the disease, 1% will test positive, which is approximately 10,000. Since only 1 person has the disease, the probability of you having the disease if you test positive is approximately 1 in 10,000.

Your chance of having that disease would be 99% only if it were the case that the base rate of people having the disease and not having the disease is equal. This is not the case here—the probability of having a disease is a million times less. Keep on the lookout for base-rate fallacies anywhere this difference in base rate is large. This is particularly relevant to security because the base rate of attacks is usually much much lower than that of normal non-malicious activity.

We can also solve this using Bayes Theorem.

$$
\begin{aligned}
D &= \textit{have disease} \\
T &= \textit{tested positive} \\
\mathbb{P}\left[D \mid T\right] &= \frac{\mathbb{P}\left[T \mid D\right]\mathbb{P}\left[D\right]}{\mathbb{P}\left[T\right]} \\
&= \frac{\mathbb{P}\left[T \mid D\right]\mathbb{P}\left[D\right]}{\mathbb{P}\left[T \mid D\right]\mathbb{P}\left[D\right] + \mathbb{P}\left[T \mid \neg D\right]\mathbb{P}\left[\neg D\right]} \\
&= \frac{0.99 * (\frac{1}{1{,}000{,}000})}{[\frac{0.99*1+0.01*999{,}999}{1{,}000{,}000}]} \\
&= \frac{0.99}{10,000.98} \\
&= \frac{1}{10,102}
\end{aligned}
$$

In mathematical terms, we can express the base-rate fallacy as making the (mis)assumption that $\mathbb{P}\left[D \mid T\right] = \mathbb{P}\left[T \mid D\right]$.