THE BUSINESS CASE FOR TLS 1.2 FOR MONITORING DATA TRANSACTIONS

Transaction security is a big deal

Transactions are the building blocks of user experience for every application, for every digital business, everywhere in the world. A transaction is a like a game of ping pong between a client and a server. It's a back and forth succession of receiving and sending data that, when successfully executed, makes up a logical unit of work such as updating a database, submitting a web form, playing a video, logging into your account, posting a Facebook message, charging a credit card, etc. If you string a bunch of these transactions together, then "voila!" you have completed a match and a fully functioning application is born.

However, the process of communication between a client and a server can be rife with peril. Hackers lurking on the sidelines can exploit security vulnerabilities between the 'acknowledge' and 'send' handshakes to hijack your transactions. This means they could masquerade as the user and seize personal information – even if no login credentials are captured. This is called a man-in-the-middle (MiM) attack.

Beware the man in the middle

So how does a MiM attack happen in the first place? It happens predominantly over unsecure

networks (like the Wi-Fi at the airport, coffee shops and college campuses) where hackers can get their hands on a users' session cookies. A MiM hacker is able to impersonate communication from both the client-side and the server-side by capturing their public keys – effectively tricking both parties into thinking they are communicating securely when in fact the messages are being relayed through the man in the middle. Once captured, the MiM hacker can manipulate the messages, get into user accounts, and wreak havoc.

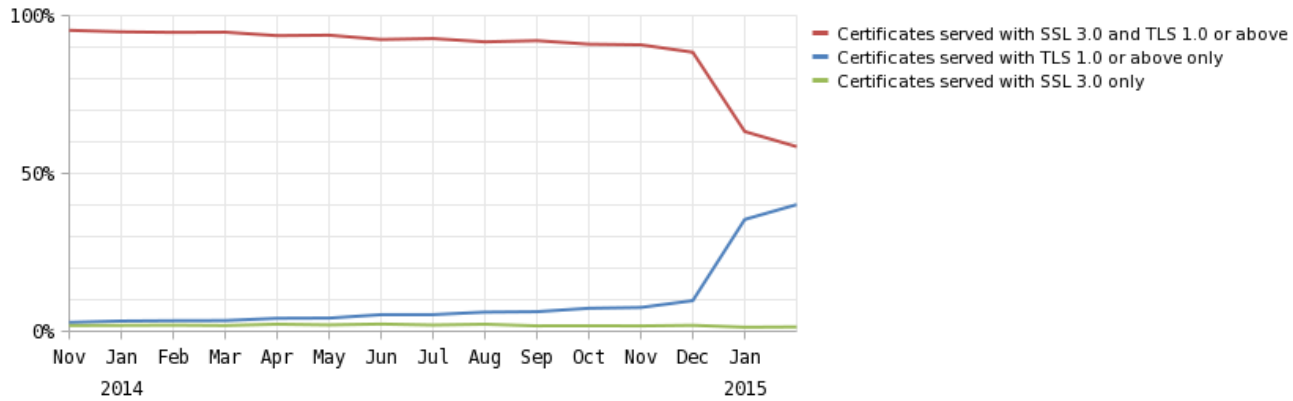
TLS, SSL and a vicious POODLE. Will we ever be safe?

To protect ourselves from the notorious MiM, we employ the use of secure channels through public key infrastructures. Transport Layer Security (TLS) and its older brother Secure Socket Layer (SSL) are such public key infrastructures used to securely communicate over HTTP/HTTPS connections. Digital certificates containing public keys are issued by a trusted third-party certificate authority (like Symantec, Go Daddy or Comodo), encrypted and exchanged between the client and the server over the network. The certificate authority is in charge of saying "Yes, this guy is who he says he is."

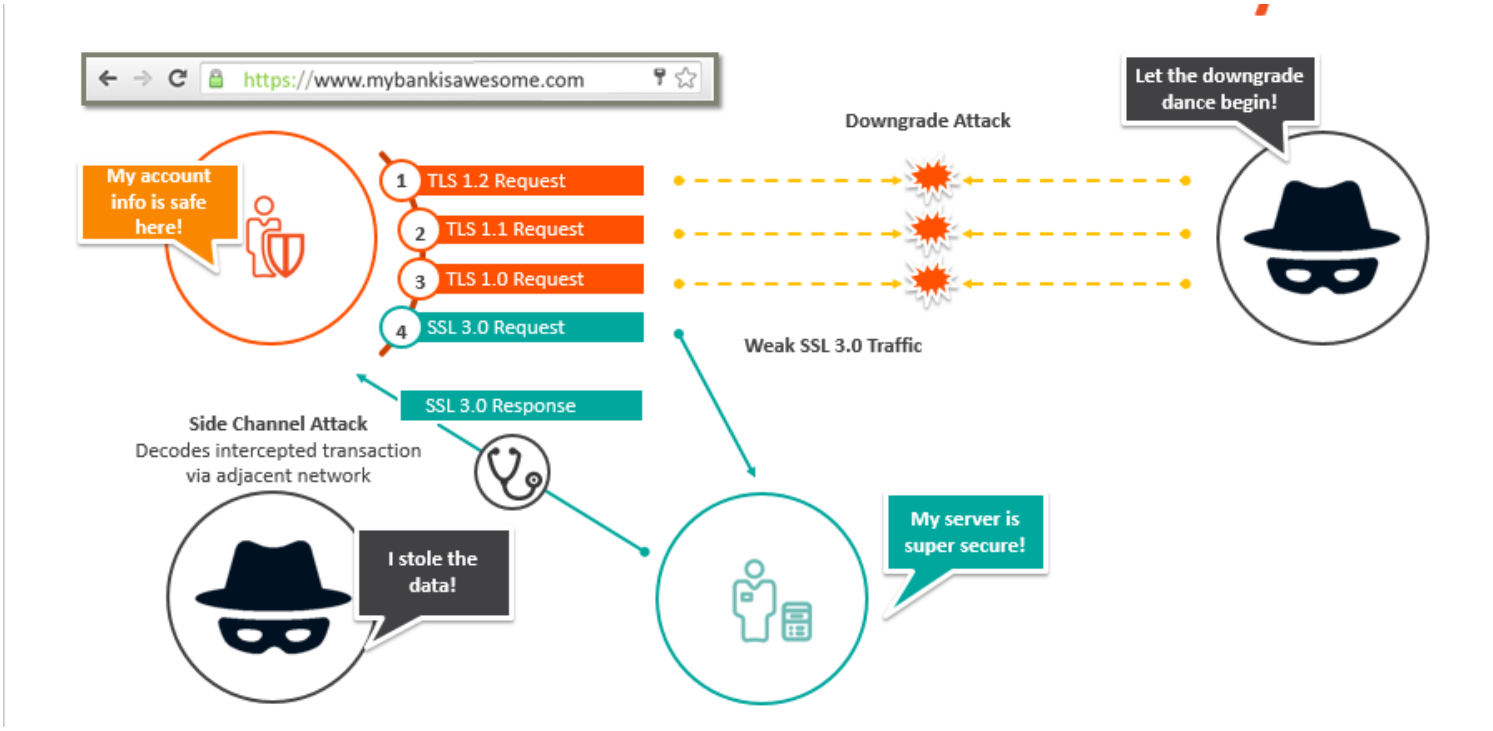
Where it becomes confusing is TLS is often referred to as its predecessor SSL – but there are some serious differences. SSL has major security holes unveiled by the POODLE (Padding Oracle on Downgrade Legacy Encryption) Bug. SSL was born in 1995, and is, in many ways, the swiss cheese of cryptographic protocols – it's holey and the ciphers it uses are easy for hackers to break.

Enter TLS; TLS is much safer. It uses symmetric key encryption that generates unique keys for every connection and negotiates the shared secret (like a password) BEFORE any data transmission happens. This way, even if a MiM is present – they won't be able to hear the secret or alter any of the data passed between the client and the server. Problem solved! Everyone move over to TLS!

By and large, everyone has. Netcraft, an organization analyzing web server, systems, hosting providers and SSL certificate authorities released a survey of security certificates issued since 2014. They found that certificate authorities issuing certificates for SSL 3.0 only are virtually null (the green line). But notice the red line – this line represents certificates issued supporting BOTH SSL 3.0 and TLS 1.0 or above. Why support both? Interoperability and backwards compatibility. Remember the ping pong game between the client and the server? It only works if they are both speaking the same language over the network, so they pass back and forth until they find the highest common denominator - lovingly referred to as the downgrade dance.



Now, in an honest transaction, the client side and the server side would agree that the highest mutual version of TLS is the clear choice. But the MiM has other ideas. If a hacker wants to hijack a transaction, and both the client and the server are backward compatible, the hacker can force the downgrade dance to begin until they are communicating over the network using SSL – at which point the hacker can exploit the less secure cipher and steal your stuff.



So how can you avoid being bit by the POODLE?

Well, the surest way is to eradicate all SSL from both the client and the server side. This is the trend, but it won't happen overnight. As the graph shows, the red line representing both SSL 3.0 and TLS 1.0 or above is dropping at the same rate that the blue line, TLS 1.0 and higher, is rising. Eventually TLS will be the only game in town. But what can we do in the meantime?

Get the TLS Fallback SCSV in place. This is basically a cipher suite that is added to the first handshake between client and server. It should be added to both the client and the server for the best protection. So when the server says, "Hello Client. I'm using TLS 1.2, can we talk?" and the client responds "I see you server. I'm using SSL 3.0 can you downgrade?" the TLS Fallback SCSV instructs the server to respond, "Sorry, I can't fallback to that version, it's not safe." At this point the communication ends.

Make sure you find all your legacy connections. From a business standpoint anything that touches the server should be brought up to date. A number of important regulatory bodies agree.

- The NIST (National Institute of Standards and Technology) require all federal organizations to use TLS. Government-only applications are required to be configured using TLS 1.1 or higher and civilian facing applications are required to be configured using TLS 1.0 or higher (for greater client-side backwards compatibility).
- The PCI (Payment Card Industry) Security Council will require TLS 1.1 or higher by June 2018 for all secure credit card transactions.
- HIPAA has security standards required for medical instrument/devices where BMC may monitor and manage the servers attached to health devices. HIPAA follows NIST guidelines for software security which requires TLS 1.0 or higher.

TLS 1.2 is quickly becoming the de facto standard for all important business data – especially in highly regulated environments. If you bucket together government, healthcare and anyone handling credit card payments you get a pretty sizeable representation of the BMC customer base.

But what about performance monitoring?

All digital businesses need to monitor the performance, availability and health of their infrastructure and applications. Full stack monitoring means that the server and applications running on it are exposed to the monitoring software. System performance monitoring tools don't generally collect medical data or PCI data directly. However, they do monitor the host or application running the business services that collect that data.

This is where it gets interesting from where I stand in the BMC Performance & Analytics group. The simplistic allegory of a ping pong match between the client and server is easy to understand – the interaction between an end-user and the application they are interfacing with.

We are like the referee of the ping pong tournament – watching all the matches, recording performance and providing general commentary about the game. For TrueSight Operations this means the agents need to be secure using the latest TLS 1.2 protocol.

Enterprise monitoring with operational analytics such as that found in the TrueSight portfolio is often reserved for business critical applications. That high level of criticality is often correlated with an elevated requirement for security. Whether the entity is monitored through an agent or agentlessly (TrueSight supports both), the confidentiality and integrity of the data generated at the monitoring entity to be transmitted over an unsecure connection for processing must be maintained. For example, a business may want to implement monitoring for a host/service running a financial application. The business then wants to send that time series data over to an IT analytics solution such as TrueSight Intelligence for further analysis. It hence becomes imperative for both the monitoring solution and the receiving analytics solution to have security standards of the highest level.

This is especially critical for customers moving their infrastructure to the cloud. Security principles change when applications are hosted in the public or hybrid cloud. Data privacy is beyond the control of any independent organization. It thus becomes an important requirement for any application data, including performance data, to be securely transmitted over the network. As an example, one of our customers is currently in the process of moving one of their key solutions to AWS. The security concern of data being communicated through end nodes (such as a monitoring agent) is top of mind for them and TrueSight Operations allows them to protect every channel with TLS 1.2 level security.

When you use TrueSight Operations, you won't get caught up in an unforeseen game of digital ping pong where thorny, up-to-no good players are in the room grabbing all the attention.

Make sure your infrastructure monitoring is safe from POODLEs.