Impact of TLS 1.3 on Enterprise Network Operations

Steve Fenter

Matt Green

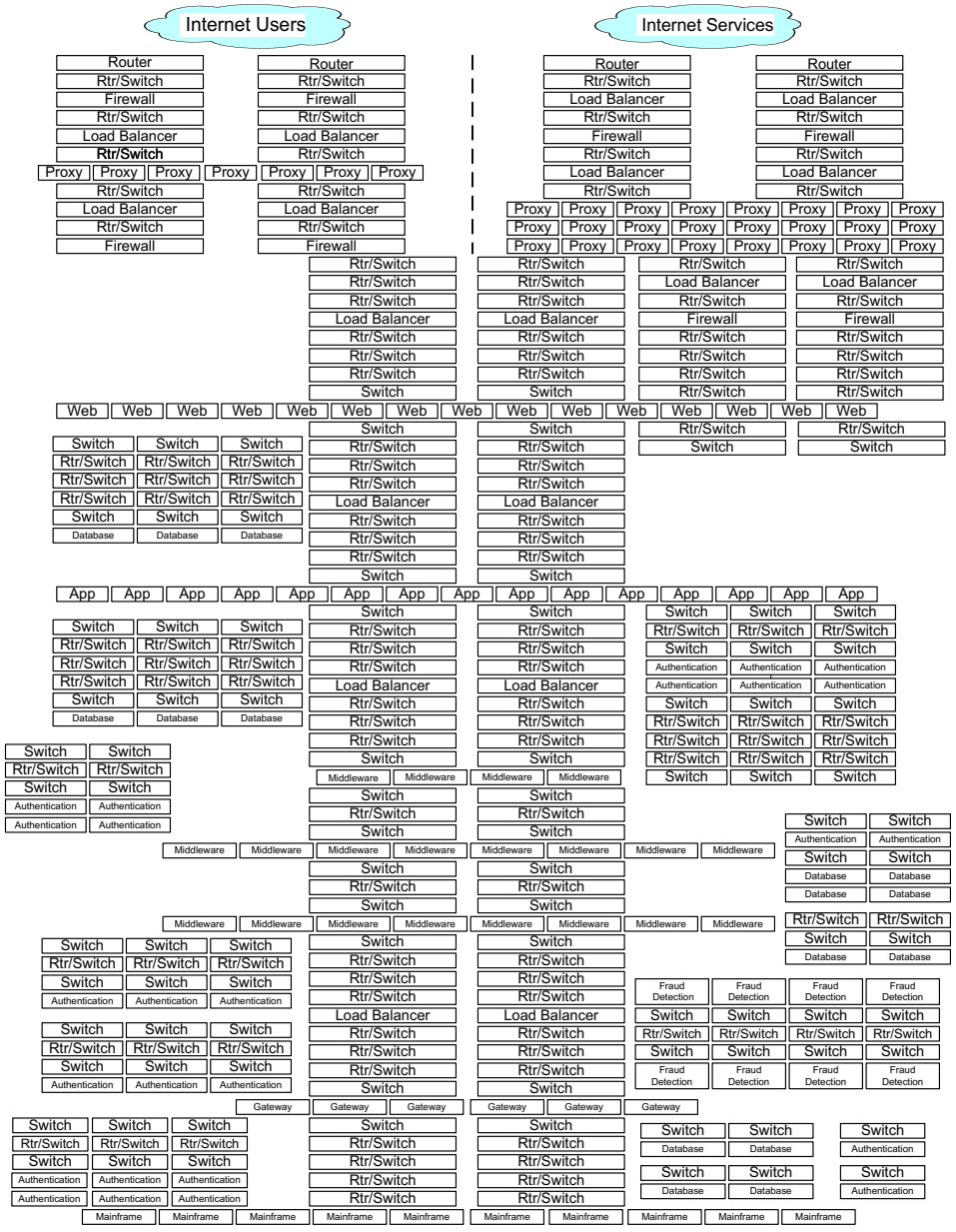
Tim Polk

Russ Housley

Enterprise Decryption Use Cases

- Wireshark PCAP decryption
- Fraud Monitoring
- IDS/IPS
- Malware Detection
- Security Incident Response
- Regulatory Requirements
- Layer 7 DDoS Protection
- NPM/APM

Enterprise Operational Support Environment



Alternative Solutions

- Proxies in the Data Center
  - Cost
  - Latency
  - Production Risk
- Endpoint Monitoring for Troubleshooting
  - Logging
  - Packet Capture with Decryption
    - Most endpoints are not built for full scale packet capture
    - Decrypted traces are needed where there is no TLS termination point

Problem Setting

- There is a need for enterprises to monitor TLS-encrypted sessions inside the datacenter
  (i.e., control of one endpoint)

- Cryptographic endpoints can always
  share cryptographic secrets (e.g., not wiretapping)

- So this is strictly an engineering problem:

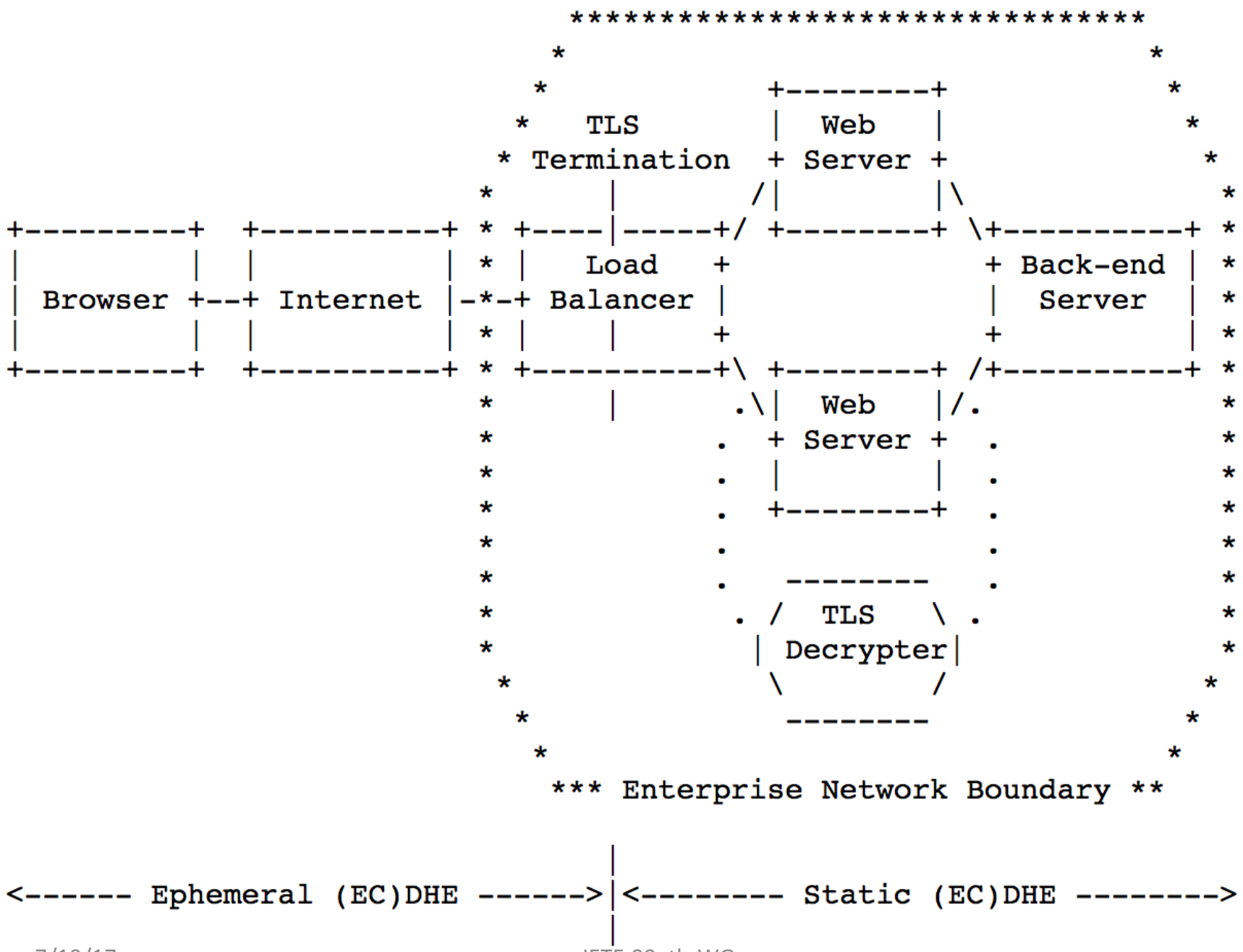
  1. How do we do this efficiently?
- 2. How do we avoid harming the TLS protocol

Possible solutions

- 1. Endpoints deliver session keys or MSes
  - Workable, but requires vast engineering and maybe too much latency for real-time monitoring
    (e.g., thousands of TLS sessions/sec)
- 2. Endpoints encode secrets in-band
  - Using an extension or encrypting into unused fields
  - Extensions may not be supported on all clients
  - Can be hard to detect
  - “Dual EC DRBG”
- 3. Endpoints use (semi)-static keys
  - No changes to TLS 1.3 protocol
  - Easy to detect
  - Reduces forward secrecy, mitigated by key rotation

Static DH Draft

- draft-green-tls-static-dh-in-tls13-01
- Proposes a server configuration for TLS 1.3
  - (Suggestion by Hugo Krawczyk)
  - To use a semi-static server key
  - (Should be rotated periodically)
  - Notes security considerations
  - Discusses key management and storage
- For use within the datacenter only
- Compatible with current TLS 1.3 draft


TLS
Termination
Web
Server
Load
Balancer
Back-end
Server
Browser
Internet
Web
Server
TLS
Decrypter
*** Enterprise Network Boundary **
<------ Ephemeral (EC)DHE ------>
<-------- Static (EC)DHE ------->

Security of Static DH

- Diffie-Hellman is well known to be secure in “ephemeral-static” mode
  - See e.g., FIPS SP 800-56A (DH-ES)
  - Similar to TLS 1.2 DHS
  - Reduces forward secrecy (well known)
  - Even in the event of a client key repeat, nonces ensure changed MS between sessions
  - TLS 1.3 standard (now) specifies EC point validation

Security of Static DH

- Diffie-Hellman is well known to be secure in “ephemeral-static” mode
  - Major concerns are implementation-specific (e.g., small subgroups)
  - TLS 1.3 draft (and FF-DH) addresses these concerns, easy to test implementations
  - This configuration does not affect most users

Harm reduction

- What is our solution space?

1. Enterprises don't adopt TLS 1.3
2. Enterprises make dramatic changes to server endpoints (e.g., deliver session keys)
3. Some really bad ideas
4. Extensions and protocol changes

- What we get from Static-DH is:

1. No significant protocol changes
2. Well-understood cryptography
3. Detectability

National Cybersecurity Center of Excellence

Increasing the adoption of standards-based cybersecurity technologies

TRANSPORT LAYER SECURITY (TLS) 1.3
VISIBILITY INSIDE THE ENTERPRISE DATA CENTER

NIST National Institute of Standards and Technology U.S. Department of Commerce

NCCoE NATIONAL CYBERSECURITY CENTER OF EXCELLENCE

NCCoE Mission

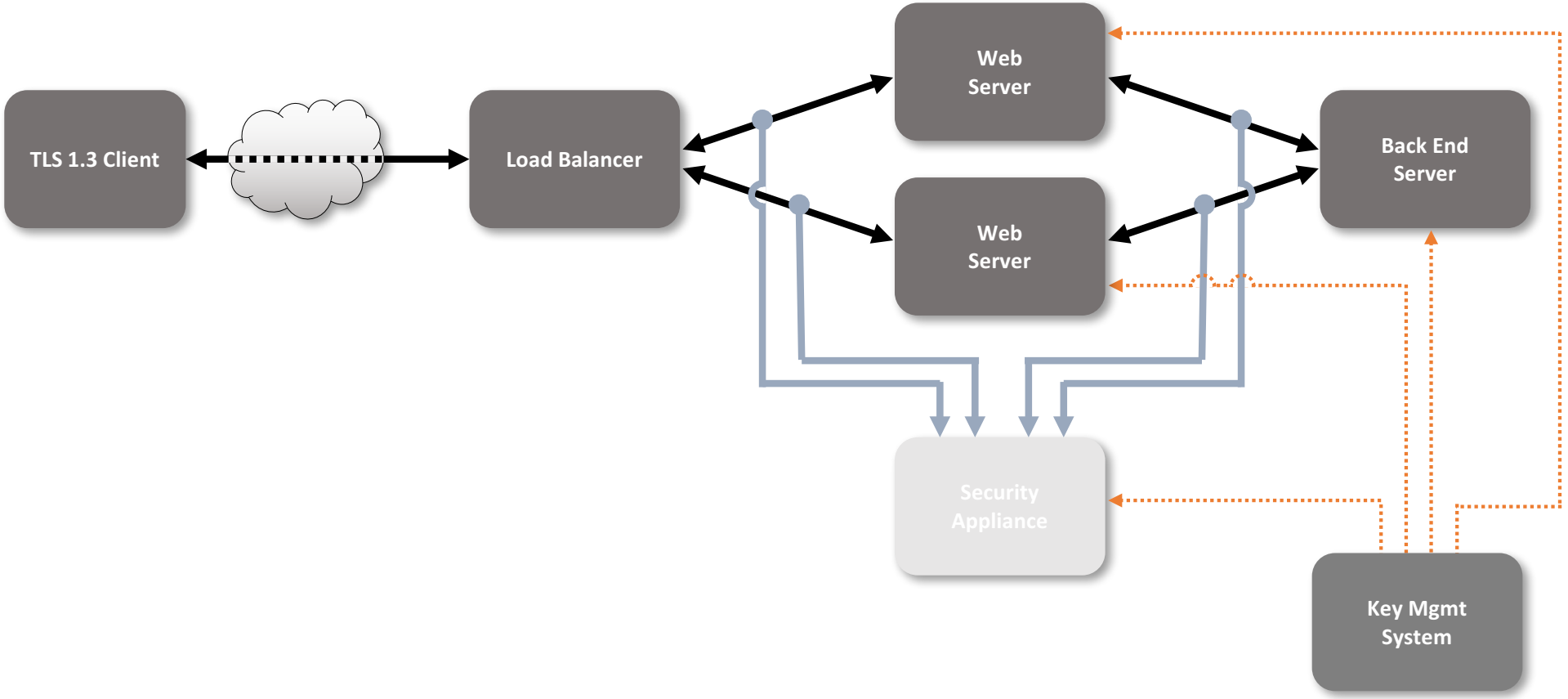
Background on TLS Visibility Project

- TLS Roundtable (May 18)
  - Included Financial, Manufacturing, Health Care and Government Sectors
  - NIST and NCEP Partners
- Roundtable and follow up discussions established that
  - Operational requirements are widely shared across all four sectors
  - Current practice shares long-lived RSA keys according to typical certificate management timelines
- By introducing a central key manager and taking advantage of automation, operational requirements can be supported while significantly enhancing security using TLS 1.3, draft-green, NIST key management guidelines, and existing standards

Objective

- Collaborate with industry and customers to demonstrate and document standardized formats and processes to support this capability
- Foster support by commercial-off-the-shelf (COTS) software and hardware vendors
- Resolve an impediment to wide deployment of TLS 1.3 across enterprise network environments, and
- Enhance security in those environments.

Notional Architecture



Four Scenarios to Implement

- Security appliance gains visibility to TLS 1.3 session plaintext between the load balancer and a web server
- Security appliance gains visibility to TLS 1.3 session plaintext between a web server and a back-end server.
- Troubleshooting tools gains visibility to TLS 1.3 session plaintext between the load balancer and a web server
- Security appliance gains visibility to TLS 1.3 session plaintext between a web server and a back-end server when session resumption is used.

What's next?

- Proposed Project Description will be published in the Federal Register and posted on the NCCoE projects page (~mid-August 2017)
  - https://nccoe.nist.gov/projects
- NCCoE will work with interested parties to refine or revise the project description (commencing September 2017)
  - May alter the architecture and protocols
  - May augment the scenarios to address additional requirements or concerns

Deliverables

- NCCoE and partners will implement the “building block” and document any decisions or software development required to achieve interoperability
  - a proof of concept implementation leveraging a standard, unmodified version of TLS 1.3 with server(s), client(s), and inspection device(s), supplemented by a centralized key management system, in a test environment that represents a typical enterprise network
  - Publish a NIST 1800 series 3-part Practice Guide
  - Submit an IETF draft describing the architecture and experiences to the ADs for consideration as an Informational RFC, preferably in coordination with the TLS WG

Proposal DOES NOT Violate the IETF Policy on Wiretapping

From RFC 2804:

Wiretapping is what occurs when information passed across the Internet from one party to one or more other parties is delivered to a third party:

1. Without the sending party knowing about the third party
2. Without any of the recipient parties knowing about the delivery to the third party
3. When the normal expectation of the sender is that the transmitted information will only be seen by the recipient parties or parties obliged to keep the information in confidence
4. When the third party acts deliberately to target the transmission of the first party, either because he is of interest, or because the second party's reception is of interest.

The server accepts the (EC)DH key from the key manager, and then uses it.

One of the parties is completely aware, and in fact enables, potential decryption by other parties.