The dark side, and the light…

Security spend is at an all time high

Security controls are at an all time high

“Interest” from BoD is at an all time high

Complexity is at an all time high
I have tons of controls. I have a smart team. So... How secure am I?

Every CISO ever
Best Intentions, But What Results?

PROXY / URL FILTER / ANTIMALWARE
FIREWALL
NGFW
IDS
EMAIL FILTER
WAF
IAM / MFA
ENDPOINT

SECURITY TEAMS
Are these controls working?
What’s the IMPACT of attack?

BOARD/EXECS/BUSINESS
Can I show security ROI?
Can I justify more investment?
Legacy Testing Methods

- **VULNERABILITY SCANNING**
  - Easy, too narrowly focused

- **PENETRATION TESTING**
  - Cumbersome, point-in-time

- **RED TEAM**
  - Detailed, creative, innovative

Automated? Comprehensive? Safe?

- Continuous
- Point-in-time

- Narrow Focus
- Full Kill Chain
Breach and Attack Simulation

**Remediate**
- Initiate automation for immediate fixes
- Send to orchestration/ticketing for ops follow-up

**Simulate Attacks**
- Cloud, network, endpoint
- Infiltration, lateral movement, exfiltration

**Prioritize Results**
- Identify and visualize where attacks are successful
- Filter and target critical issues for actionable results
Simulating the Adversary

Continuously validate security with thousands of attacks — safely

- **Infiltration**: Simulated malware drops, Executable files, Registry entries
- **Lateral Moves**: Brute force, Remote code execution, Pass-the-hash
- **Exfiltration**: Sample data, Direct methods, Covert methods

Blocked Methods
Successful Methods
Data Exfiltration Pathways
Recommended Actions
Breach and Attack Simulation

Validate your controls— with the same techniques attackers use

▸ Get more from existing security by optimizing config and ensuring controls work in concert
▸ Minimize security exposure due to human error, updates, and policy changes
▸ Prepare for audits by validating segmentation and other compliance controls
▸ Test alerting and action plans for SOC or MSSP teams, and provide breach scenario training
▸ Get business rationalization for security investment, prove security against headline attacks

Validate your defenses before the attackers do
Simulating the Adversary

Do, or do not. There is no try.

3,400 breach methods executed
11.5 Million simulations
Across verticals and deployment sizes
Simulating the Adversary: Results

- Malware manages to evade perimeter defenses
- Encrypted files not scanned
- Leaving it up to the endpoint

### Top Infiltration Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Success Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>WannaCry 2.0 Ransomware</td>
<td>63.4%</td>
</tr>
<tr>
<td>EXE packed inside a JavaScript</td>
<td>60.9%</td>
</tr>
<tr>
<td>Carbanak/Anunak HTTP Malware Transfer</td>
<td>59.8%</td>
</tr>
<tr>
<td>EXE inside a VBS using HTTP</td>
<td>56.5%</td>
</tr>
<tr>
<td>EXE inside a CHM File</td>
<td>55.9%</td>
</tr>
</tbody>
</table>
Simulating the Adversary: Results

- Lateral moves looked like infiltration
- LAN trust is too high
- Is internal traffic safer than Internet traffic?

**Top Lateral Movement Methods**

<table>
<thead>
<tr>
<th>Method</th>
<th>Success Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Malware Transfer - Petya worm via HTTP/S</td>
<td>69.4%</td>
</tr>
<tr>
<td>EternalRocks - Transfer via HTTP/S</td>
<td>68.9%</td>
</tr>
<tr>
<td>EXE inside WSF (as XML) using HTTP</td>
<td>67.3%</td>
</tr>
<tr>
<td>EXE inside JAR using HTTP</td>
<td>67.0%</td>
</tr>
<tr>
<td>Lazarus Buffer Transfer</td>
<td>66.5%</td>
</tr>
</tbody>
</table>
Simulating the Adversary: Remediation

- Dramatically increased security in three weeks
- No new investment
- Conflicting rules, misconfiguration, underutilization
Top Considerations: Breach and Attack Simulation

- **Safe** for production deployment
- **Continuous** validation – not point in time
- **Actionable**, prioritized findings
- **Complete** kill chain, modular, “black box”

**Safe | Continuous | Actionable | Thorough**
SIMULATE ATTACKS
VALIDATE CONTROLS
HARNESS THE HACKER

SafeBreach