infoblox

SOLUTION BRIEF

NEXT-GEN FIREWALL + BLOXONE® THREAT DEFENSE — A POWERFUL COMBINATION

Enhance defense-in-depth of both solutions through seamless integration

THE CHALLENGE

Network and security teams rely on NGFWs to strengthen their organization's overall security posture by defining network boundaries and applying security policy rules that permit or block network traffic and application access. In today's dynamic threat environment, however, NGFWs and other perimeter defenses can become overwhelmed by the task of scrutinizing network traffic for potential threats. NGFWs are also not designed to see certain pervasive threats that rely on DNS pathways.

TOP NGFW PLATFORMS INTEGRATE WITH INFOBLOX

FORTINET

paloalto NETWORKS

CHECK POINT

IMPROVE NGFW PERFORMANCE AND EFFICIENCY WITH INTEGRATED DNS DETECTION AND RESPONSE

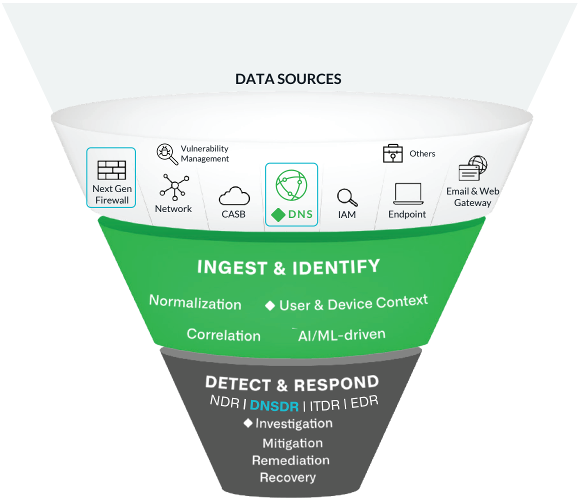
The DNS protocol plays a central role in all network communications. It is also inherently insecure, which is why it is the vector for more than 90 percent of malware and is also invoked in pervasive ransomware, data exfiltration and distributed denial of service attacks. DNS Detection and Response (DNSDR) expands the defense-in-depth of NGFWs by identifying and remediating DNS threats that elude other security measures.

The industry's leading DNSDR solution, Infoblox BloxOne® Threat Defense, integrates with preeminent NGFW platforms, enhancing the capabilities of each solution and elevating overall SecOps effectiveness. It delivers these benefits by:

- Using DNS as a central point of control for on-premises, cloud, and/or remote, work-at-home and BYOD personal devices
- Reducing security burdens by blocking bad traffic earlier in the attack cycle and at the first point of query—before they even reach NGFWs
- Providing proactive protection of 100% of DNS traffic from accessing lookalike domains, connecting to malicious domains and exfiltrating data
- Using DNS context to enrich existing security orchestration, automation and response (SOAR) capabilities
- Tracking who, where and what devices are on your network, currently and historically, via context-rich endpoint visibility

DNSDR ENHANCES NEXT-GEN FIREWALLS AND THE ENTIRE XDR ECOSYSTEM

DNS Visibility and NGFWs Can Work Together in the XDR Framework



DNSDR and NGFW solutions are essential components of extended detection and response (XDR) solutions. DNSDR implementations like BloxOne Threat Defense improve the performance and efficiency of not only NGFWs, but of core XDR capabilities across the security ecosystem. Through APIs and pervasive automation, BloxOne Threat Defense shares indicators of compromise (IoCs) and other verified threat intelligence with other tools and systems, enabling SecOps to reduce Mean Time to Respond (MTTR) for fast-moving cyber threats.

"We deployed BloxOne Threat Defense to block threats within the DNS architecture. Shortly after that, [our cloud NGFW vendor] called to tell us the firewall was seeing 70% less malicious activity and there might be a problem."

- SecOps Director at a US State Agency on how the ways BloxOne Threat Defense reduces NGFW burdens

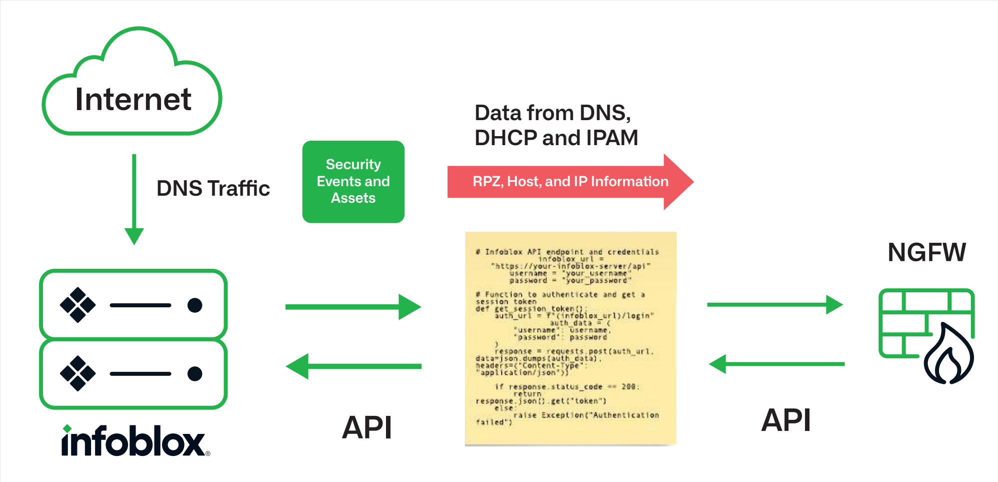
"Infoblox really gives our teams a 'confidence uplift.' This really lowers the stress on our team."

- Venture Capital Customer

TOP 10 REASONS CUSTOMERS CHOOSE BLOXONE THREAT DEFENSE

1. Accelerate Time to Value
2. Detect Threats Other Solutions Miss
3. Achieve Anywhere, Hybrid Visibility and Control
4. Stop Attacks Earlier in the Attack Chain
5. Boost SecOps Efficiency
6. Speed Investigation and Response by 3X
7. Unlock the power of DNS Threat Intel
8. Optimize the Security Ecosystem
9. Get More from Security Investments
10. Gain Greater Context by Merging IPAM with DNS

HOW INFOBLOX AND NGFWs INTERACT


Internet
DNS Traffic
Security Events and Assets
Data from DNS, DHCP and IPAM
RPZ, Host, and IP Information
NGFW
infoblox.
API
API

```
# Infoblox API endpoint and credentials
                infoblox_url =
  "https://your-infoblox-server/api"
        username = "your_username"
        password = "your_password"

# Function to authenticate and get a
session token
def get_session_token():
    auth_url = f"{infoblox_url}/login"
                auth_data = {
        "username": username,
        "password": password
    }
    response = requests.post(auth_url,
data=json.dumps(auth_data),
headers={"Content-Type":
"application/json"})

    if response.status_code == 200:
        return
response.json().get("token")
    else:
        raise Exception("Authentication
failed")
```

To learn more about the benefits of our Cybersecurity Ecosystem visit https://www.infoblox.com/products/cybersecurity-ecosystem/

Corporate Headquarters
2390 Mission College Blvd, Ste. 501
Santa Clara, CA 95054