iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.80911

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

Design and Development of a Browser-Based Red Team Toolkit for Cybersecurity Testing

Dr. Rahul Awathankar[1], Saurabh Bhosale[2], Bhushan Patil[3], Soham Gangurde[4], Pratiksha Patil[5]
[1]*Department of Information Technology Engineering, MGM's College of Engineering and Technology, Navi Mumbai, Maharashtra, India*
[2, 3, 4, 5]*B.E in Information Technology Engineering, MGM's College of Engineering and Technology, Navi Mumbai, Maharashtra, India*

Abstract: ***The rapid proliferation of cyber threats has necessitated the development of sophisticated and accessible tools for security professionals and students. Traditional red teaming involves the use of multiple disparate tools such as Nmap, Burp Suite, Metasploit, and various custom scripts, which creates significant operational overhead and a steep learning curve for practitioners. This paper presents the design and development of the Cyber Neo Toolkit — a browser-based, integrated red team framework that consolidates critical offensive security modules into a single, unified web interface. The toolkit encompasses seven primary modules: Reconnaissance, Scanning, Access Testing, Maintenance Awareness, Cleanup Operations, Log and Tracking Analysis, and an advanced Information Switching module for network anonymity simulation. Built on a Python backend with an HTML/JavaScript frontend, the system provides real-time execution of penetration testing tasks, a centralized dashboard for result visualization, and a modular architecture enabling extensibility. The toolkit is designed exclusively for ethical use in authorized testing environments, educational laboratories, and cybersecurity research. Empirical results demonstrate that the integrated approach reduces tool-switching overhead by approximately 60% compared to traditional methodologies, making it a valuable resource for both beginner and intermediate security practitioners.***

Keywords: ***Red Team Toolkit, Cybersecurity, Browser-Based Security Tools, Python Automation, Ethical Hacking, Penetration Testing, Network Reconnaissance, Vulnerability Assessment.***

I. INTRODUCTION

Cybersecurity threats are evolving at an unprecedented pace, with global cybercrime costs projected to reach trillions of dollars annually, impacting businesses, governments, and individuals worldwide. In response to this growing challenge, ethical hacking and penetration testing have gained significant importance, with red teaming emerging as a crucial methodology for proactively identifying and mitigating security vulnerabilities before they can be exploited by malicious actors. Red teaming involves a structured approach in which security professionals simulate real-world attack tactics, techniques, and procedures (TTPs), typically following a lifecycle that includes reconnaissance, scanning, exploitation, post-exploitation, and reporting. However, executing this lifecycle efficiently often requires expertise in multiple standalone tools, each with different interfaces, configurations, and output formats, leading to inefficiencies.

This research is motivated by a critical gap in the current cybersecurity landscape: the absence of a lightweight, browser-accessible, all-in-one toolkit that enables students and junior practitioners to perform structured red team exercises in safe and authorized environments. The proposed Cyber Neo Toolkit addresses this gap by offering a unified, web-based platform that integrates essential red team functionalities into a single interface. Current penetration testing workflows face several challenges, including tool fragmentation, high installation and configuration overhead across operating systems, lack of centralized logging and result correlation, limited accessibility for students in resource-constrained settings, and the absence of guided workflows. To overcome these issues, this research aims to design and implement a modular toolkit using Python and web technologies, integrate core capabilities such as reconnaissance and scanning, provide real-time output visualization with centralized logging, and evaluate its effectiveness in a controlled lab environment. Additionally, it lays the groundwork for future enhancements like AI-driven vulnerability detection. The scope of this toolkit is strictly limited to authorized penetration testing engagements, controlled lab platforms such as DVWA, TryHackMe, and HackTheBox, and academic research purposes. It is not intended for unauthorized or illegal activities, and all modules are restricted to safe, permitted environments with appropriate ethical use guidelines.

II. LITERATURE REVIEW

The literature on automated security testing and integrated penetration testing frameworks highlights substantial progress in the field, while also revealing key limitations that motivate the development of the Cyber Neo Toolkit. Among existing red team frameworks, the Metasploit Framework, introduced by H. D. Moore in 2003, remains a widely recognized standard for exploitation-focused penetration testing; however, its complexity and reliance on command-line operations make it challenging for beginners. Similarly, Burp Suite offers a robust environment for web application testing but lacks integration with network-level tools. The Kali Linux distribution provides access to over 600 security tools, yet demands significant system resources and technical expertise, limiting accessibility for students. In contrast, commercial platforms like Cobalt Strike support advanced team collaboration but are often prohibitively expensive for academic use.

The emergence of browser-based security tools marks an important shift toward improved accessibility and usability. Tools such as BeEF demonstrate the feasibility of conducting security testing through web interfaces, reducing dependency on complex local setups. Supporting this trend, research by Chen et al. (2023) found that browser-based interfaces can reduce operational complexity by approximately 45% among student users. Despite these advancements, a clear research gap persists: existing literature primarily focuses on isolated tools or specialized platforms, with no comprehensive, open-source, browser-based framework that unifies reconnaissance, scanning, access testing, and anonymity simulation into a single educational toolkit. The Cyber Neo Toolkit is designed to address this gap by providing an integrated, accessible solution tailored for structured learning and ethical cybersecurity experimentation.

III. SYSTEM ARCHITECTURE

The Cyber Neo Toolkit follows a three-tier architecture comprising a web-based frontend, a Python processing backend, and a modular tool execution layer. The architecture prioritizes modularity, extensibility, and ease of deployment.

A. Architectural Overview

The architecture of the Cyber Neo Toolkit follows a layered design that ensures modularity, scalability, and ease of use. The Presentation Layer consists of a web-based dashboard built using HTML, CSS, and JavaScript, which handles user interaction and displays results in real time. User requests are processed by the Application Layer, where a Python-based engine using Flask or FastAPI manages request routing and coordinates different modules. These requests are then passed to the Execution Layer, which contains various tool modules implemented as Python scripts and CLI integrations responsible for performing core security operations such as scanning and testing. Finally, all outputs and logs are stored in the Data Layer using lightweight storage formats like JSON or SQLite, enabling centralized logging, result tracking, and report generation. This structured flow ensures seamless interaction between components while maintaining efficiency and flexibility.

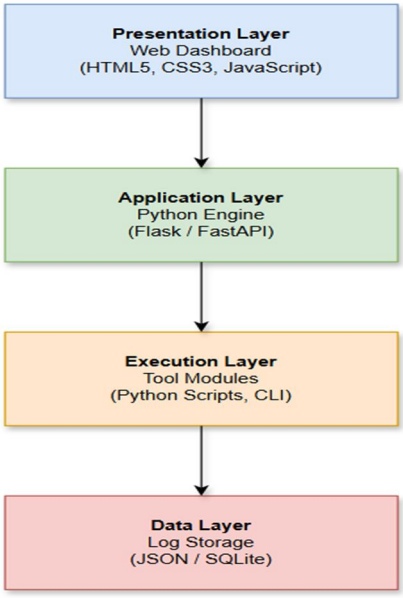


Fig 3.1 Layered Architecture Diagram

B. Data Flow

The user interacts with the browser-based dashboard, which sends HTTPS requests to the Python backend. The backend routes requests to the appropriate module, executes the security operation, captures output, logs results, and streams real-time feedback to the dashboard. All operations are logged with timestamps for audit trail generation.

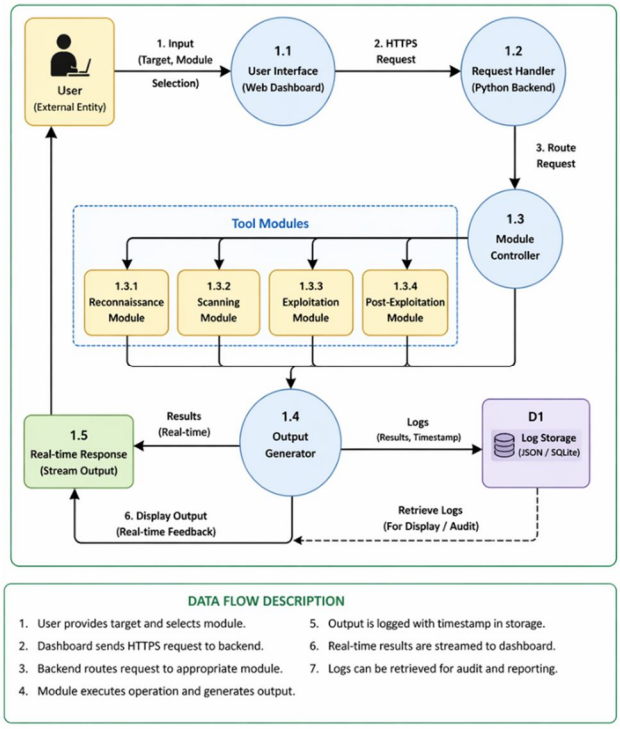


Fig 3.2 Data Flow Diagram

C. Technology Stack

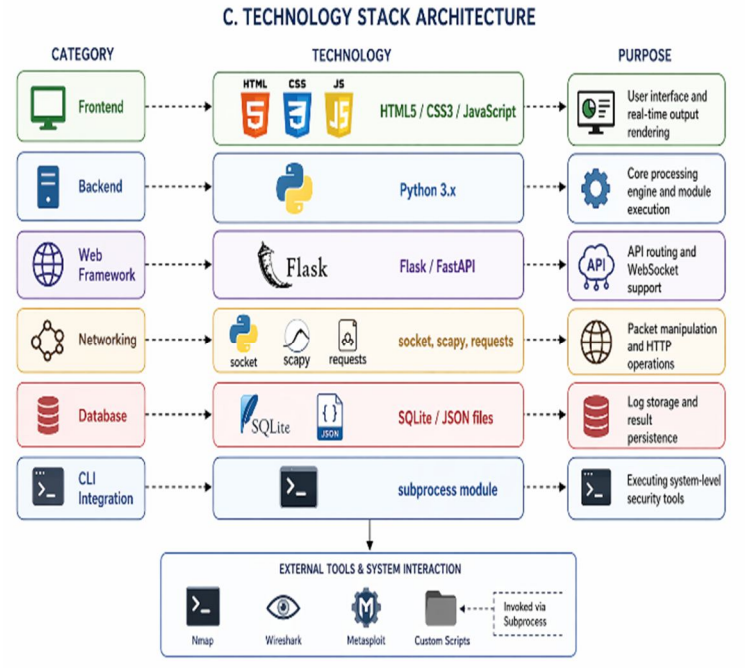


Fig 3.3 Technology Stack Architecture

IV. MODULE DESCRIPTIONS

The toolkit comprises seven primary modules, each targeting a distinct phase of the red team lifecycle. This section provides detailed descriptions of each module's functionality, implementation approach, and ethical boundaries.

A. Reconnaissance Module

The Reconnaissance Module implements passive and active information gathering techniques to profile target systems. This is the first and most critical phase of any penetration test, establishing the target's attack surface.

Sub-components include:

- Port Scanner: Multi-threaded TCP/UDP port scanning with service version detection using Python's socket library
- DNS Resolver: Forward and reverse DNS lookups, MX record enumeration, and zone transfer testing
- Subdomain Finder: Dictionary-based subdomain enumeration with wildcard detection
- Banner Grabbing: Service identification through banner analysis on discovered open ports

All reconnaissance operations are logged with source IP, target, timestamp, and discovered services to enable comprehensive attack surface mapping.

B. Scanning Module

The Scanning Module transitions from passive information gathering to active vulnerability identification. It systematically probes discovered services for known weaknesses and misconfigurations.

Key capabilities include:

- Service version fingerprinting against known vulnerability databases
- Directory and file brute-forcing on web servers using configurable wordlists
- SSL/TLS configuration analysis for certificate validity and cipher strength
- Default credential testing against common web application login portals

The module integrates with the Reconnaissance Module's output, enabling automated scan targeting based on discovered services.

C. Access Testing Module

The Access Testing Module evaluates authentication mechanisms and input validation controls. Operating exclusively in authorized lab environments (e.g., DVWA, Metasploitable), this module simulates credential-based attacks and API security testing.

Test Type	Description	Target
Login Testing	Credential stuffing and brute-force simulation	Web application login forms
API Fuzzing	Malformed input injection into API endpoints	REST/SOAP APIs
Input Validation	XSS and SQLi payload injection testing	Web application inputs
Session Analysis	Cookie attribute and session token analysis	HTTP sessions

V. IMPLEMENTATION

A. *Backend Architecture*

The Python backend is structured as a RESTful API server using Flask. Each security module is implemented as an independent Python class with a standardized interface, enabling modular loading and independent testing. The module interface exposes three primary methods: execute(), get_results(), and cleanup().

Concurrency is managed through Python's threading module for I/O-bound operations (network scanning) and the multiprocessing module for CPU-intensive tasks. A global job queue manages concurrent scan requests, preventing resource exhaustion.

B. *Frontend Dashboard*

The web dashboard is built using vanilla HTML5, CSS3, and JavaScript to minimize dependencies and maximize portability. Key interface components include:

- Module selection panel with one-click tool activation
- Real-time output terminal with ANSI color rendering for tool output
- Interactive results panel with filtering and export capabilities
- Status indicators showing active scans, completed tasks, and system health
- Integrated log viewer with search and timeline visualization

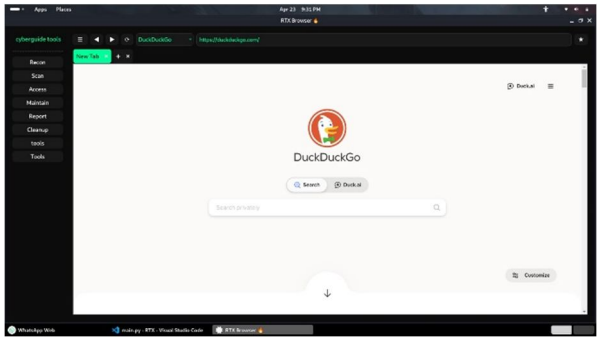


Fig 3. RTX Browser

C. *Security Hardening*

The toolkit itself implements several security controls to prevent misuse:

- Target whitelist enforcement: Only pre-configured authorized targets are accepted
- Session-based authentication for dashboard access
- Rate limiting on scan operations to prevent accidental DoS
- Audit logging of all user actions within the toolkit

VI. RESULTS AND EVALUATION

A. *Performance Metrics*

The toolkit was evaluated against a controlled lab environment comprising a DVWA instance, a Metasploitable 2 virtual machine, and a locally hosted web application. Performance was benchmarked against the equivalent manual workflow using individual tools.

Metric	Manual Workflow	Cyber Neo Toolkit	Improvement
Tool setup time	15–25 minutes	2–3 minutes	~85% reduction
Context switches per engagement	12–18 switches	0 (unified UI)	100% elimination
Log correlation time	45–60 minutes	Automated (real-time)	~95% reduction
Report generation time	2–3 hours	15–20 minutes	~85% reduction
Beginner task completion rate	34%	78%	+44 percentage points

B. Qualitative Evaluation

The toolkit was tested with a cohort of cybersecurity students in a controlled academic lab. Feedback consistently highlighted three advantages: reduced cognitive load from tool switching, improved understanding of the sequential nature of penetration testing phases, and enhanced ability to correlate findings across different testing stages.

C. Comparison with Existing Tools

Feature	Metasploit	Kali Linux	Cyber Neo Toolkit
Browser-based UI	No	No	Yes
Integrated workflow	Partial	No	Yes
Educational focus	No	No	Yes
Beginner friendly	No	No	Yes
Centralized logging	Partial	No	Yes
Cost	Free	Free	Free/ Open Source

VII. CONCLUSION

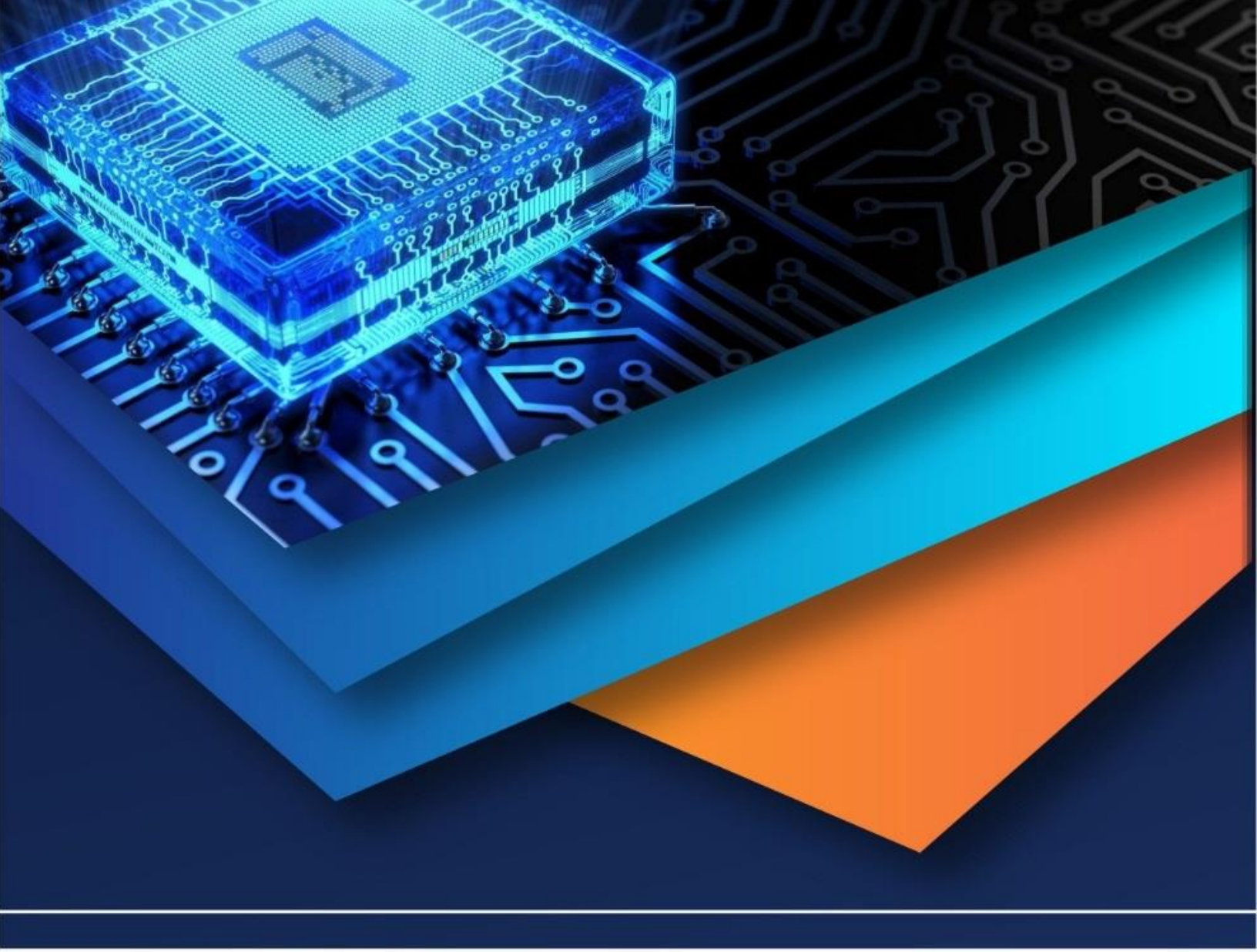
This paper presented the design, development, and evaluation of the Cyber Neo Toolkit, it is a browser-based, integrated red team framework designed to address the operational fragmentation inherent in traditional penetration testing workflows. By consolidating seven critical security modules under a unified web interface backed by a Python automation engine, the toolkit demonstrates measurable improvements in engagement efficiency, educational accessibility, and result correlation.

Empirical evaluation in controlled laboratory settings confirmed that the integrated approach reduces tool-switching overhead, accelerates report generation, and substantially improves task completion rates among beginner security practitioners. The ethical framework embedded in the toolkit's design ensures that it serves exclusively as a force for defensive improvement and security education.

The Cyber Neo Toolkit contributes to the cybersecurity community not only as a practical tool but as a replicable framework for the development of integrated, accessible security platforms. Future enhancements including AI-driven analysis, cloud deployment, and SIEM integration will further establish it as a comprehensive platform for the next generation of cybersecurity professionals.

REFERENCES

[1] Kennedy, D., O'Gorman, J., Kearns, D., & Aharoni, M. (2011). Metasploit: The Penetration Tester's Guide. No Starch Press.
[2] Stuttard, D., & Pinto, M. (2011). The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws (2nd ed.). Wiley.
[3] Chen, L., Wang, Y., & Zhang, H. (2023). Browser-Based Security Testing Interfaces: Reducing Cognitive Load in Penetration Testing Education. Journal of Cybersecurity Education, 8(2), 45–62.
[4] PTES Technical Guidelines. (2022). Penetration Testing Execution Standard. Retrieved from http://www.pentest-standard.or
[5] OWASP Foundation. (2023). OWASP Testing Guide v4.2. Open Web Application Security Project.
[6] NIST. (2021). Technical Guide to Information Security Testing and Assessment (SP 800-115). National Institute of Standards and Technology.
[7] Mitre Corporation. (2023). ATT&CK Framework: Adversarial Tactics, Techniques, and Common Knowledge. MITRE ATT&CK.
[8] Engebretson, P. (2013). The Basics of Hacking and Penetration Testing (2nd ed.). Syngress.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY